

POR

J. DECONCHAT

E.P.S.I.

EDICIONES ELISA, S. A.

PROGRAMAS

102
PROGRAMAS PARA
ZX81
Y SPECTRUM

JACQUES
DECONCHAT

102
PROGRAMAS PARA
ZX81 Y SPECTRUM



EDICIONES ELISA, S.A.

1984

Título original de la obra: 102 PROGRAMMES POUR
SINCLAIR ZX ET TIMEX

Primera edición: noviembre 1984.

ISBN: 84-7622-001-4.
Depósito legal: B. 35.291-1984.

Printed in Spain Impreso en España

Gráficas SOCITRA. Arquímedes, 3. HOSPITALET DE LLOBREGAT.

Índice

Prólogo

En este libro encontrará más de 100 programas, redactados en Basic Sinclair, escritos y presentados cada vez en dos versiones: una para el ZX 81 1 K octeto (utilizable casi siempre sobre el ZX 80 New Rom, sin modificación) y la otra para el ZX·Spectrum (16 K en la versión de base).

El hecho de presentar solamente programas muy cortos (menos de 1 K octeto en todos los programas de la versión ZX 81) se debe a dos hechos: por una parte, el deseo de guiar al lector hacia un aprendizaje progresivo del lenguaje, utilizando para ello juegos muy cortos, pero, sin embargo, bastante completos y logrados; por otra parte, la intención de no caer en la clásica trampa de los programas demasiado largos, muy difíciles de introducir en la máquina sin error y que resultan más cómodos de tener en cassettes *pregrabados que, por otro lado, se encuentran cada vez más corrientemente en el comercio.*

Cada nivel de programa (hay 5) está precedido de un corto estudio de las instrucciones Basic que se emplean en los juegos que siguen. Se exponen las diferencias entre el ZX 81 y el ZX Spectrum, así como las diferencias con los otros Basic.

Además, los programas que se ofrecen están "abiertos"; es decir, han sido concebidos, dentro de lo que cabe, para ser fácilmente utilizados y modificados si el tamaño de la memoria disponible es superior: por ejemplo, un juego como Othello, que sólo puede practicarse en un tablero 6 × 6 en la versión de 1 K, podrá practicarse sin ninguna modificación en un tablero de 8 × 8 o 10 × 10 en una versión de 16 K o más (todos los juegos pueden utilizarse en 16 K sin modificación).

Al final de cada programa, se facilitan indicaciones para sugerir posibles mejoras o modificaciones, para hacer el juego más atractivo, más logrado o más completo.

Organización de la obra

Los juegos están clasificados por dificultad creciente, siguiendo 5 niveles:

- **El nivel 1** permite estudiar las instrucciones elementales del lenguaje Basic: **LET, PRINT, INPUT, IF — THEN —, FOR ..., NEXT, GOTO, GOSUB.**
- **En el nivel 2**, se utilizarán las instrucciones de confección y manejo de los elementos de una tabla: **DIM, R(I).**
- **En el nivel 3**, se recurre a la potente instrucción de tratamiento de series de caracteres: **R$(TO)**, y a las instrucciones gráficas: **PLOT, UNPLOT.**
- **En el nivel 4**, se encuentran las instrucciones propias de los juegos interactivos: **INKEY$** y **SCROLL.**
- **En el nivel 5**, por último, se descubren las instrucciones fronterizas del Basic, que permiten la manipulación directa en la memoria del aparato: **PEEK** y **POKE.**

Cada juego se presenta de una manera muy sistemática, mediante una breve descripción, seguida del estudio detallado de cada una de las líneas importantes del programa. A continuación, se dan las indicaciones necesarias para el buen funcionamiento del juego, así como algunos elementos para posibles extensiones o mejoras del juego propuesto.

Los programas **ZX 81** y **ZX Spectrum** se presentan uno al lado del otro, y la numeración de las líneas es, dentro de lo que cabe, la misma en las dos versiones. El interés de esta fórmula es múltiple: debería permitir a los antiguos usuarios del ZX 81, que deseen pasarse al ZX Spectrum, descubrir rápidamente las semejanzas y diferencias (mínimas) entre los dos aparatos; esto permitirá, por otro lado, aclarar cualquier duda de interpretación en la lectura de un programa (por ejemplo, impresión un poco borrosa) y entender mejor el significado de los códigos que se emplean en la versión ZX 81 para disminuir la ocupación de la memoria.

Para aumentar la facilidad de transposición de los programas del ZX 81 al ZX Spectrum, este último se ha utilizado sistemáticamente en posición C A P S L O C K (bloqueo de las teclas en posición de mayúscula); de esta manera, se suprimen voluntariamente las minúsculas que existen en el Spectrum; será fácil modificar los programas para utilizarlas.

Por otra parte, existen algunas diferencias notables entre el Basic del ZX 81 y el del ZX Spectrum, que son generalmente fáciles de evitar; la lectura comparada de los programas permitirá encontrar fácilmente los diversos métodos que existen para poner remedio a este problema.

DIFERENCIAS MÁS NOTABLES

- Existen dos modalidades, FAST y SLOW, en el ZX 81, y ninguna en el Spectrum. No hay ningún inconveniente práctico, a no ser la necesidad de añadir en algunos programas una contemporización para disminuir la velocidad del Spectrum, que funciona según la modalidad SLOW del 81, pero a la velocidad de la modalidad FAST.

- Existe una instrucción SCROLL en el ZX 81. Se soluciona en el Spectrum anulando el SCROLL ?, que aparece automáticamente si la pantalla está llena y se provoca la impresión de un carácter de retorno del carro (CHR$(13)) en la línea 21: se escribirá PRINT AT 2Ø,Ø:POKE 23692,Ø:PRINT CHR$(13).
- La posibilidad de escribir varias instrucciones por línea en el ZX Spectrum, separadas por 2 puntos.
- La posibilidad de imprimir un texto con la ayuda de una instrucción INPUT y de esperar algunas variables en el Spectrum.
- La posibilidad de leer directamente el carácter que se encuentra en la posición X,Y de la pantalla en el Spectrum; la instrucción

  SCREEN$(X,Y) sustituirá así las dos líneas:

  PRINT AT X,Y;

  LET C$ = CHR$ PEEK(PEEK 16398 + 256 × PEEK 16399)

  del ZX 81.

  (Para el color, se utilizará la instrucción ATTR, que es de un manejo más delicado.)
- Cambio de los códigos de los distintos caracteres: el ZX Spectrum utiliza el código ASCII, lo que no sucedía en el ZX 81.
- Algunos caracteres gráficos del ZX 81 no están ya disponibles en el Spectrum (pueden reemplazarse por el color).
- Las instrucciones **READ, RESTORE** y **DATA,** que no existían en el ZX 81, están presentes en el ZX Spectrum (en esta obra se utilizan muy raramente, con el fin de respetar la analogía con el ZX 81).
- La instrucción **PLOT** del ZX Spectrum permite direccionar 256 × 176 puntos, en lugar de los 64 × 44 del ZX 81, lo que permite un gráfico de alta resolución, pero complica la traducción de algunos programas.
- La instrucción **UNPLOT** se reemplaza en el Spectrum por **PLOT OVER1.**

Existen otras muchas diferencias, pero prácticamente no tendremos la ocasión de encontrarlas en el curso de este estudio. Parece importante precisar aquí que las analogías son mucho más numerosas que las diferencias y que los programas escritos para el ZX 81 serán, salvo casos muy particulares, fácilmente adaptables al ZX Spectrum; en cambio, la velocidad de carga y de protección de los programas en *cassette* no es la misma, y los programas *cassette* del ZX 81 no se utilizarán directamente en el Spectrum.

Se han utilizado algunas técnicas para limitar la ocupación de la memoria, sobre todo en la versión ZX 81. Cuando estas técnicas recurren a las propiedades particulares del Basic Sinclair (a menudo es el caso), se utilizan directamente en el ZX Spectrum, y se descubrirán con interés estudiando los programas propuestos (utilización de funciones lógicas para suprimir comprobaciones, utilización del GOTO y del GOSUB calculados, etc.). En cambio, en algunas ocasiones, el procedimiento utilizado, si economiza memoria, hace que el programa sea más difícil de interpretar: en este caso, la codificación no se ha retenido en la versión Spectrum, lo que permitirá comprender el sentido de los símbolos utilizados dando una simple ojeada en la versión del ZX Spectrum. Así, en el ZX 81, Ø se sustituye por NOT PI, 1 por SGN PI y 3 por INT PI; esta técnica, aunque también es rentable en el Spectrum, no se ha considerado por las razones que se indican más arriba. Asimismo, una gran cantidad de números se sustituyen por los códigos de caracteres correspondientes en el ZX 81. Así, GOTO 2ØØ se escribe frecuentemente GOTO CODE "COS", PRINT AT 16, 15 se escribirá PRINT AT CODE"(",CODE"?", etc.

NIVEL 1

N.º	*Nombre del programa*	*Tema y particularidades*
1	**El número misterioso**	Juego de deducción para un jugador a varios niveles.
2	**Cubo**	Juego de dibujo: perspectiva sobre una pantalla.
3	**Caliente, frío**	Lógico y de azar para un jugador; varios niveles.
4	**Las cerillas**	Juego de reflexión contra el ZX.
5	**Cerillas para varios**	Juego de reflexión y de estrategia para varios jugadores.
6	**Calendario perpetuo**	Confección de un calendario según mes y año.
7	**Ruleta**	Juego de azar para varios jugadores; simulación.
8	**Batalla naval**	Encontrar un solo barco, con algunas particularidades.
9	**Golf**	Un recorrido de 18 hoyos, para un solo jugador.
10	**Solfeo**	Aprender solfeo con la ayuda del ZX.
11	**El juego del 21**	Juego de azar contra el ZX.
12	**Par-impar**	Un juego de casino en tres tiempos (uno o varios jugadores).
13	**Simón**	Repetir una secuencia de cifras cada vez más larga.
14	**Adivinar una cifra**	Juego de casino para uno o varios jugadores.
15	**El juego de Wytthof**	Un juego de reflexión muy simple para dos jugadores.
16	**421**	Una simulación simplificada para varios jugadores.
17	**Rebotes**	Un juego de destreza muy visual para un jugador.
18	**Jackpot**	Juego de azar muy clásico (un jugador).
19	**El juego del haba**	Un juego de reflexión para dos jugadores.
20	**Juego de la oca aleatorio**	Un viejo juego al estilo ZX para dos jugadores.

Instrucciones elementales

Aquí sólo se consideran las instrucciones fundamentales o las instrucciones de programación; otras muchas sentencias o funciones particulares se utilizarán si se presenta el caso; por ejemplo:

- **CLS para borrar la pantalla,**
- **PAUSE para parar momentáneamente el desarrollo del programa,**
- **RND para obtener un número aleatorio entre Ø y 1,**
- **INT para tomar la parte completa del resultado,**

etcétera.

Las instrucciones fundamentales de la programación, que se utilizan desde el nivel I y que se presentan prácticamente en todos los programas, son:

LET; PRINT; INPUT; GOTO; IF — THEN — ;FOR ... NEXT; GOSUB ... (RETURN);

solamente siete instrucciones de base, que van a examinarse a continuación más detalladamente.

LA INSTRUCCIÓN LET O INSTRUCCIÓN DE ASIGNACIÓN

Es la instrucción por excelencia de los lenguajes de programación evolucionados, ya que su papel principal es permitir al programador una gestión transparente de la memoria; esto significa, en la práctica, que para almacenar una cantidad en la memoria, por ejemplo 12, no es necesario saber en qué lugar de la memoria va a depositarse; es suficiente con asignarle un nombre autorizado, y por este nombre la cantidad será más tarde encontrada y manejada.

- **Si la cantidad es numérica,** el nombre será una simple letra del alfabeto, o un conjunto de cifras o letras que empiecen obligatoriamente por una letra.
- **Si la cantidad es alfanumérica,** se aplica la misma regla para el nombre que, sin embargo, deberá ir seguido del símbolo $, y la variable se escribirá entre comillas.

Ejemplo: LET A1 = 12; la variable numérica A1 toma el valor 12.
LET NOM$ = "RENOIR"; la variable numérica NOM$ toma el valor RENOIR.

En la mayoría de los Basic, la instrucción LET puede omitirse, lo que facilita la escritura: A = 12; B$ = "TOTO", etc.

Una vez realizada la asignación, el microordenador sabrá cada vez que le hablen de A1 que se trata del número 12 y si se le habla de NOM$, del nombre RENOIR, etc.

LA INSTRUCCIÓN PRINT O INSTRUCCIÓN DE SALIDA (PANTALLA)

Es una instrucción de diálogo que permite a la máquina visualizar en la pantalla un texto, el resultado de un cálculo o una variable; es el único medio que posee el microordenador para dar a conocer sus resultados. Así, PRINT A1 provoca la visualización de 12 (si anteriormente se ha efectuado LET A1 = 12) y PRINT NOM$ la visualización de RENOIR (si se ha efectuado LET NOM$ = "RENOIR"); si se utiliza después de borrar la pantalla (CLS), PRINT provoca la visualización de la 1.ª línea de arriba a la izquierda.

Se utilizan diversos parámetros para indicar al ZX dónde debe visualizar sus resultados o sus símbolos (y comprendidos los dibujos).

- **PRINT TAB N** provoca una visualización en la línea de impresión actual, que comienza en la columna N, a partir de la izquierda (Ø ≤ N ≤ 31).
- **PRINT AT X,Y** provoca una visualización que empieza en la línea X (a partir de arriba, con Ø ≤ X ≤ 19) y en la columna Y (a partir de la izquierda, con Ø ≤ Y ≤ 31).
- **Un ; (punto y coma)** al final de PRINT permite encadenar otra impresión sin repetir la instrucción PRINT o indicar que la próxima impresión será, si no se indica lo contrario, realizada exactamente a continuación de la precedente.
- **La , (coma)** provoca una tabulación predefinida automática.

LA INSTRUCCIÓN INPUT O INSTRUCCIÓN DE ENTRADA (TECLADO)

Esta instrucción permitirá introducir y asignar directamente en la variable que se indica por el INPUT un número o un texto, realizado en modalidad conversacional, a medida de las necesidades; así, cuando el ZX encuentra una instrucción INPUT, se para y devuelve el control. Puede entonces introducirse una información apropiada y señalar el fin de esta entrada con la ayuda de la tecla **NEWLINE**, que devolverá el control al ZX.

Por ejemplo, si en el programa el ZX encuentra INPUT B, se para, y deberá teclearse un número B cualquiera seguido de NEWLINE (si se teclea un carácter distinto de una cifra o un punto decimal, se obtendrá un mensaje de error: es conveniente prever la visualización de un texto antes de la parada, indicando lo que tiene que introducirse).

En otros aparatos, así como en el Spectrum, existen otros medios de introducir informaciones (LET, DATA, etc.), pero éstas no tienen la flexibilidad del INPUT, ya que entonces la información queda plasmada en el programa y difícilmente puede modificarse.

En el ZX 81, la instrucción INPUT está muy simplificada: no puede mencionarse por un texto el tipo de la información, ni introducir varias informaciones, unas después de otras. En el ZX Spectrum la instrucción INPUT es parecida a la INPUT de los otros BASIC, que permiten visualizar un texto e introducir sucesivamente diversas variables, en caso necesario.

LA INSTRUCCIÓN GOTO O LA INSTRUCCIÓN DE BIFURCACIÓN INCONDICIONAL

Una de las particularidades del lenguaje Basic, que lo hace fácilmente accesible, es que ***cada línea del programa debe ir precedida de un número;*** estas líneas se ejecutan a continuación en secuencias por el microordenador, en numeración creciente. Por varias razones, puede ser necesario, durante la ejecución de un programa, no seguir este orden secuencial de forma muy rígida; entonces se utiliza la ***instrucción GOTO, que pemite desviar la ejecución del programa de su curso normal,*** con la posibilidad, sin embargo, de volver a

tomarlo a continuación por otro GOTO. El Basic, en el caso del Basic Sinclair, por lo general sólo permite las bifurcaciones a un número de línea conocido (GOTO 2Ø, por ejemplo, o GOTO 2ØØ, etc.).

Una de las particularidades más destacables del Basic Sinclair es que también permite el GOTO calculado, lo que significa que, detrás de la instrucción GOTO, puede ponerse una variable o también una expresión aritmética o lógica de cálculo; lo que hace de ella una instrucción de bifurcación condicional muy potente, parecida al CASE OF del Pascal; puede escribirse,

Por ejemplo:

GOTO(1ØØ AND I = 1) + (2ØØ AND I = 2) + (3ØØ AND I = 3) o GOTO I × 1ØØ, que envía a las líneas 1ØØ, 2ØØ o 3ØØ, según el valor de I.

LA INSTRUCCIÓN IF ⎵ THEN ⎵ , O INSTRUCCIÓN DE COMPROBACIÓN

La instrucción **IF ⎵ THEN ⎵** , muy potente en el Basic Sinclair, puede utilizarse de distintas maneras:

- En rotura de secuencia condicional, bajo la forma **IF ⌊ condición = verdadera ⌋ THEN GOTO ⌊ nnn ⌋ número de línea.** Es el uso más corriente de esta comprobación, que de todas formas podrá mejorarse por el hecho de que la condición puede ser una expresión aritmética o lógica compleja, evaluada lógicamente y que puede ser calculada por el mismo GOTO.
- También podrá utilizarse la comprobación bajo la forma: **IF ⌊ condición ⌋ THEN ⌊ instrucción ⌋** ; la instrucción siguiente a THEN puede ser una asignación LET, una impresión PRINT, una entrada INPUT o también otra comprobación IF ⎵ THEN ⎵ ; puede igualmente escribirse:

Por ejemplo: IF A THEN IF B THEN PRINT A + B.

LA INSTRUCCIÓN FOR ... NEXT, O LA INSTRUCCIÓN DE BUCLES

A menudo, en programación querrá ejecutarse un número entero de veces una cierta serie de instrucciones (por ejemplo, hacer avanzar un móvil desde la casilla Ø hasta la casilla 2Ø, etcétera.). Más que utilizar una comprobación y una bifurcación, es preferible utilizar un ***bucle;*** la variable de bucle toma valores enteros o decimales y el paso de esta variable (cantidad en que aumenta o disminuye a cada paso) puede también tomar valores enteros o decimales.

Así, si se escribe:

```
FOR I = Ø TO 2Ø STEP 2
   |  |   Instrucciones
NEXT I
```

Las instrucciones que contiene el bucle se ejecutarán por I = Ø, por I = 2 y hasta que I = 2Ø, es decir 11 veces. La variable I puede utilizarse para realizar un cálculo en el bucle, pero no deberá modificarse dentro de él. Puede salir del bucle antes de que I = 2Ø mediante una comprobación sobre una cantidad cualquiera.

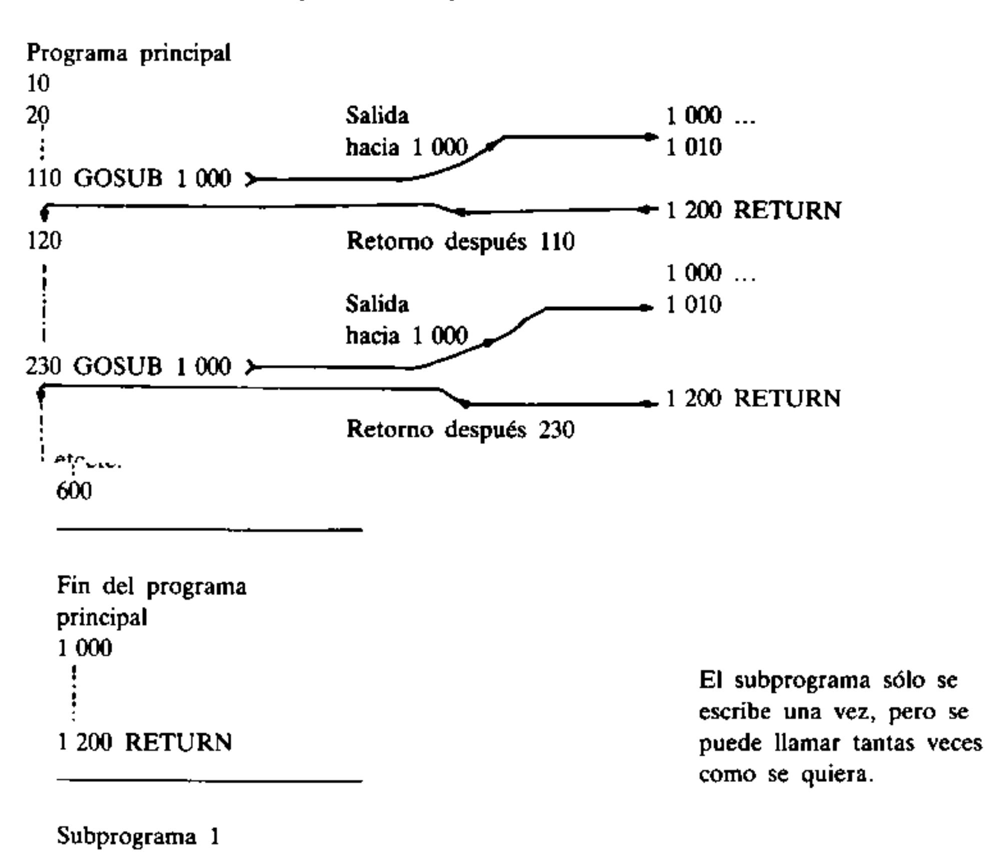
Si no se ha precisado ***el paso del bucle (STEP),*** se tomará automáticamente el valor 1.

El Basic del ZX permite colocar un número indefinido de bucles (limitado por la memoria disponible).

LA INSTRUCCIÓN GOSUB O INSTRUCCIÓN DE LLAMADA A SUBPROGRAMAS

Cuando la misma serie de instrucciones tenga que ejecutarse varias veces y llamarse desde varios sitios del programa, más que escribirla varias veces se intentará redactarla una sola vez y se la situará fuera del programa propiamente dicho (por ejemplo, se le dará un número de línea de inicio superior al número de la última línea del programa); independientemente de las dificultades que puedan encontrarse a nivel de los nombres de las variables, cuando quiera utilizarse esta secuencia, bastará con llamarla con **la instrucción GOSUB número de línea.** El programa "efectuará una bifurcación" a la parte del programa llamado, aunque no podrá volver y seguir su desarrollo normal, a menos que encuentren en dicho programa una **instrucción RETURN,** o instrucción de retorno.

A continuación, se especifica el esquema teórico de su funcionamiento.



Un subprograma podrá eventualmente llamar a otro subprograma, y así sucesivamente, sin otra limitación, en el Basic Sinclair, que el tamaño de la memoria.

Es importante precisar que, en el Basic Sinclair, al igual que en el GOTO, el GOSUB también puede calcularse, lo que permitirá una gran concisión y flexibilidad de la programación.

Por ejemplo: puede escribirse GOSUB A × 1 ØØØ, y si A = 1, se ejecutará el subprograma 1 ØØØ; si A = 2, el programa 2 ØØØ, etc.

Los Basic corrientes realizan esto mediante la instrucción ON A GOSUB 1 ØØØ, 2 ØØØ, – – –, que no tiene, ni mucho menos, la misma flexibilidad.

El número misterioso

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: juego de deducción para un solo jugador.
Dificultad del juego: fácil.

EL JUEGO: Se pide al ordenador que confeccione un número de n cifras (n deberá estar comprendido entre 1 y 7).

A continuación se intenta descubrirlo, con el mínimo de intentos posibles, recibiendo solamente la información MÁS GRANDE o MÁS PEQUEÑO.

EL PROGRAMA: La línea 4Ø permite confeccionar el número que ha de adivinarse.

Las entradas se realizan por las líneas 6Ø a 9Ø.

La línea 1ØØ comprueba si el resultado es exacto.

La línea 11Ø señala un intento demasiado pequeño y la línea 12Ø un intento demasiado grande.

El contador de intentos está en la línea 13Ø.

Las líneas 2ØØ a 24Ø se ocupan de la visualización del resultado y de la gestión de la partida siguiente.

PARA JUGAR: Se pone en marcha el programa con RUN.

Se obtiene una primera visualización ¿NÚMERO DE CIFRAS?, a la que se responde con un número entero entre 1 y 7, seguido de NEWLINE.

Se obtiene entonces la visualización de INTENTO N.° 1, seguido de la pregunta SU NÚMERO.

Se introduce un número y se pulsa NEWLINE.

Se obtiene entonces una respuesta DEMASIADO PEQUEÑO o DEMASIADO GRANDE, o si se ha descubierto el número, una visualización HA ENCONTRADO ... EN ... INTENTOS. Si el resultado no es exacto, se vuelve a INTENTO N.° 2 y SU NÚMERO.

Al final de la partida, se obtiene la pregunta OTRA PARTIDA ¿(S o N)?; se responde S para jugar otra partida (seguido de NEWLINE).

Puede pararse en cualquier momento introduciendo una letra en respuesta a la pregunta SU NÚMERO.

POSIBLES EXTENSIONES: El jugador podrá también adivinar otra cosa que no sea una cifra (por ejemplo, una palabra, utilizando el orden alfabético o incluso otra cosa).

EL NÚMERO MISTERIOSO

PROGRAMA ZX 81

```
  5 CLS
 10 PRINT AT 3,1;"NUMERO DE CIF
RAS :";
 20 INPUT N
 25 IF N>7 THEN RUN
 30 PRINT N
 40 LET X=INT (RND*10**N+1)
 50 LET J=1
 60 PRINT AT 6,1;"INTENTO NO ";J
 70 PRINT AT 8,1;"SU NUMERO :";
 80 INPUT M
 90 PRINT M
100 IF M=X THEN GO TO 200
110 IF M<X THEN PRINT AT 10,1;"
DEMASIADO PEQUENO"
120 IF M>X THEN PRINT AT 10,1;"
DEMASIADO GRANDE"
130 LET J=J+1
140 PAUSE 120
150 PRINT AT 8,12;"
           "
160 PRINT AT 10,1;"
           "
170 GO TO 60
200 CLS
210 PRINT AT 8,1;"HA ENCONTRADO
";X;" EN ";J;" INTENTOS"
220 PRINT AT 12,1;"OTRA PARTIDA
(S O N)?"
230 INPUT R$
240 IF R$="S" THEN RUN
```

```
NUMERO DE CIFRAS :4

INTENTO NO 4

SU NUMERO :600

DEMASIADO PEQUENO
```

PROGRAMA ZX SPECTRUM

```
  5 CLS
 10 PRINT AT 3,1;"NUMERO DE CIF
RAS :";
 20 INPUT N
 25 IF N>7 THEN RUN
 30 PRINT N
 40 LET X=INT (RND*10↑N+1)
 50 LET J=1
 60 PRINT AT 6,1;"INTENTO NO ";
J
 70 PRINT AT 8,1;"SU NUMERO :";
 80 INPUT M
 90 PRINT M
100 IF M=X THEN GO TO 200
110 IF M<X THEN PRINT PAPER 2;
INK 5;AT 10,1;"DEMASIADO PEQUENO
"; BEEP 1.3,-10
120 IF M>X THEN PRINT PAPER 5;
INK 2;AT 10,1;"DEMASIADO GRANDE"
; BEEP 1.3,10
130 LET J=J+1
140 PAUSE 120
150 PRINT AT 8,14;"
           "
160 PRINT AT 10,1;"
           "
170 GO TO 60
200 CLS
210 PRINT AT 8,1;"HA ENCONTRADO
";X;" EN ";J;" INTENTOS"
220 PRINT FLASH 1; INVERSE 1; I
NK 4;AT 12,1;"OTRA PARTIDA (S O
N)?"
230 INPUT R$
240 IF R$="S" THEN RUN
```

```
NUMERO DE CIFRAS :3

INTENTO NO 2

SU NUMERO :50

DEMASIADO PEQUENO
```

Cubo

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: ninguna.
Categoría del juego: dibujo.
Dificultad del juego: ninguna.

EL JUEGO: El programa propuesto se limita a realizar, con la ayuda de instrucciones PRINT, una vista de un cubo en perspectiva caballera.

Podría, por ejemplo, incorporarse a un juego basado en la capacidad de reconocimiento de las formas geométricas elementales.

EL PROGRAMA: El dibujo del cubo se realiza por la impresión de puntos, utilizando dos bucles: el bucle I (1Ø-5Ø) y el bucle I (6Ø-9Ø).

El nombre del objeto representado se visualiza en la línea 1ØØ (aquí: CUBO).

La línea 11Ø es prácticamente inútil, ya que el programa se para cuando no hay más instrucciones para ejecutar.

PARA JUGAR: Es suficiente con poner en marcha la ejecución con RUN y mirarla.

Puede pulsarse cualquier tecla para salir del programa.

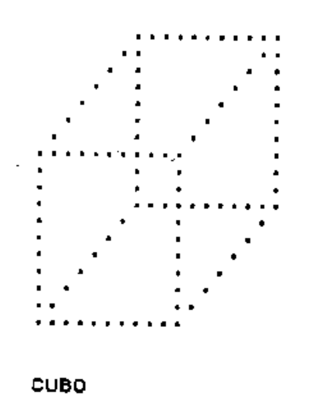
POSIBLES EXTENSIONES: La instrucción PLOT permitirá obtener un trazado más preciso.

Puede imaginarse un juego utilizando paralelepípedos rectángulos de distintos tamaños, por ejemplo, con la visualización de 3 lados, y habría que encontrar su volumen, etc.

CUBO

PROGRAMA ZX 81

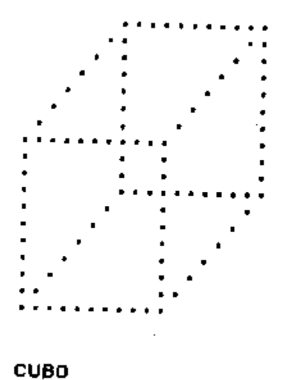
```
  5 LET L=7
  7 LET O=NOT PI
 10 FOR I=0 TO 9
 20 PRINT AT O,L+I;".";AT L,I;"
.";AT 10,L+I;".";AT 17,I;"."
 30 PRINT AT L+I,O;".";TAB 10;"
."
 40 PRINT AT I,L;".";TAB 17;"."
 50 NEXT I
 60 FOR I=0 TO L
 70 PRINT AT L-I,I;".";TAB 10+I
;"."
 80 PRINT AT 17-I,I;".";TAB 10+
I;"."
 90 NEXT I
100 PRINT AT 21,O;"CUBO"
110 PAUSE 4E4
```



CUBO

PROGRAMA ZX SPECTRUM

```
  5 LET L=7
  7 LET O=0
 10 FOR I=0 TO 9
 20 PRINT AT O,L+I;".";AT L,I;"
.";AT 10,L+I;".";AT 17,I;"."
 30 PRINT AT L+I,O;".";AT L+I,1
0;"."
 40 PRINT AT I,L;".";AT I,17;".
"
 50 NEXT I
 60 FOR I=0 TO L
 70 PRINT AT L-I,I;".";AT L-I,1
0+I;"."
 80 PRINT AT 17-I,I;".";AT 17-I
,10+I;"."
 90 NEXT I
100 PRINT AT 21,O;"CUBO"
110 PAUSE 4E4
```



CUBO

Caliente, frío

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: mediana.
Categoría del juego: solitario.
Dificultad del juego: mediana.

EL JUEGO: Es uno de los juegos más antiguos del mundo que aún se practica; sin embargo, esta vez es el ordenador el que hace de pareja: el ordenador va a esconder un objeto en un determinado sitio de una reja, cuyas dimensiones escoge uno mismo. Para encontrarlo, sólo dará algunas indicaciones bajo la forma de CALIENTE o FRÍO, etc.

Hay que descubrir el objeto escondido con el mínimo de intentos posibles (el resultado se visualizará al final de la partida).

EL PROGRAMA: Las líneas 7 y 8 definen la posición del objeto.

La línea 11Ø calcula la distancia, en relación con dos coordenadas que se han introducido en las líneas 7Ø y 9Ø.

La línea 12Ø comprueba si se ha ganado.

Las líneas 14Ø y 15Ø normalizan la distancia (que debe tener un sentido, cualesquiera que sean las dimensiones de la reja inicial).

La línea 16Ø efectúa la salida, si existe, hacia uno de los programas de visualización.

PARA JUGAR: Se pone en marcha el programa con la orden RUN; después se introduce el tamaño de la reja (cuadrada) y se pulsa NEWLINE.

A partir de la siguiente visualización (DAR X), se introduce la supuesta coordenada X, seguida de NEWLINE.

Se obtiene como respuesta un texto que es variable según se esté más o menos lejos del objeto que se busca (esto va desde COMPLETAMENTE HELADO a QUEMANDO). Cuando se encuentra, se obtiene la visualización del número de intentos realizados.

Para salir del programa antes del final de la partida, basta con dar una respuesta alfanumérica a una de las preguntas DAR... (teclear A y NEWLINE, por ejemplo).

POSIBLES EXTENSIONES: Si se dispone del color, será interesante encontrar una forma más visual de simbolizar la distancia, por ejemplo, un pavimento de baldosas que vayan del rojo al azul, en los lugares que se prueban.

También puede realizarse en un espacio de tres dimensiones, esconder varios objetos, etc.

CALIENTE, FRÍO

PROGRAMA ZX 81

```
   1 LET F=7
   2 LET G=11
   3 LET S=NOT PI
   4 PRINT "TAMANO DE LA REJA ";
   5 INPUT D
   6 PRINT D
   7 LET X=INT (RND*D+1)
   8 LET Y=INT (RND*D+1)
  11 PAUSE 120
  15 CLS
  20 PRINT AT INT PI,SGN PI;"DAR
 X ";
  70 INPUT A
  80 PRINT A;AT 5,SGN PI;"DAR Y
 ";
  90 INPUT B
 100 PRINT B
 105 LET S=S+SGN PI
 110 LET T=ABS (X-A)+ABS (Y-B)
 120 IF NOT T THEN GO TO 6000
 130 PRINT AT F,SGN PI;"ESTA "
 140 LET N=INT (SQR D)/T+1
 150 IF N>5 THEN LET N=5
 160 IF T<=(SQR D)+2 THEN GO TO
N*1000
 170 PRINT AT F,G;"COMPLETAMENTE
 HELADO"
 180 GO TO G
1000 PRINT AT F,G;"CASI TEMPLADO
"
1010 GO TO G
2000 PRINT AT F,G;"TEMPLADO"
2010 GO TO G
3000 PRINT AT F,G;"CALIENTE"
3010 GO TO G
4000 PRINT AT F,G;"MUY CALIENTE"
4010 GO TO G
5000 PRINT AT F,G;"QUEMANDO"
5010 GO TO G
6000 CLS
6010 PRINT "ENCONTRADO EN ";S;"
INTENTOS"
```

```
DAR X 3

DAR Y 3

ESTA       TEMPLADO
```

PROGRAMA ZX SPECTRUM

```
   1 LET F=7
   2 LET G=11
   3 LET S=0
   4 PRINT "TAMANO DE LA REJA ";
   5 INPUT D
   6 PRINT D
   7 LET X=INT (RND*D+1)
   8 LET Y=INT (RND*D+1)
  11 PAUSE 120
  15 CLS
  20 PRINT AT 3,1;"DAR X ";
  70 INPUT A
  80 PRINT A;AT 5,1;"DAR Y ";
  90 INPUT B
 100 PRINT B
 105 LET S=S+1
 110 LET T=ABS (X-A)+ABS (Y-B)
 120 IF NOT T THEN GO TO 6000
 130 PRINT AT F,1;"ESTA "
 140 LET N=INT (SQR D)/T+1
 150 IF N>5 THEN LET N=5
 160 IF T<=(SQR D)+2 THEN GO TO
N*1000
 170 PRINT PAPER 1; BRIGHT 1;AT
F,G;"COMPLETAMENTE HELADO"
 180 GO TO G
1000 PRINT PAPER 5;AT F,G;"CASI
TEMPLADO"
1010 GO TO G
2000 PRINT PAPER 4;AT F,G;"TEMPL
ADO"
2010 GO TO G
3000 PRINT PAPER 6;AT F,G;"CALIE
NTE"
3010 GO TO G
4000 PRINT PAPER 3; BRIGHT 1;AT
F,G;"MUY CALIENTE"
4010 GO TO G
5000 PRINT PAPER 2;AT F,G;"QUEMA
NDO"
5010 GO TO G
6000 CLS : BEEP 0.1,1: BEEP 0.1,
1: BEEP 0.1,1: BEEP 0.1,1: BEEP
0.1,1
6010 PRINT "ENCONTRADO EN ";S;"
INTENTOS"
```

```
DAR X 2

DAR Y 2

ESTA       TEMPLADO
```

Las cerillas

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: mediana.
Dificultad del juego: mediana.

EL JUEGO: Al principio de la partida, el jugador deberá escoger una cierta cantidad de cerillas y decidir la toma máxima autorizada. Después se decide quién empieza la partida, si el jugador o el ZX.

Cada cual coge por turno tantas cerillas como quiera, dentro del límite del número máximo autorizado. El que coge la última cerilla se considera el perdedor.

EL PROGRAMA: El juego del jugador empieza en la línea 11Ø.

El del ZX empieza en la línea 2ØØ.

La línea 13Ø controla la validez de la jugada del jugador.

La línea 14Ø administra el cambio de jugador.

La jugada del ZX se calcula en un bucle no formal, de la línea 21Ø a la línea 24Ø.

El resultado se visualiza en la línea 1Ø2Ø.

Si la partida ha terminado, la línea 1Ø4Ø lo detecta y bifurca a la línea 2ØØØ para el mensaje final.

PARA JUGAR: Empieza la ejecución con RUN, y ante la visualización del mensaje TOTAL se introduce el número de cerillas, seguido de NEWLINE. Después se introduce la toma máxima autorizada, seguida de NEWLINE. Si el jugador desea empezar, teclea a continuación Ø; si no, teclea 1 y a continuación NEWLINE.

Después de cada jugada, se visualiza el número de cerillas que quedan, así como el ZX SU TURNO cuando le toca coger al jugador.

El que toma la última cerilla pierde.

POSIBLES EXTENSIONES: Podrá considerarse una visualización gráfica en color más espectacular, dejar al azar escoger el número de cerillas y la toma máxima, etc.

LAS CERILLAS

PROGRAMA ZX 81

```
  10 PRINT "TOTAL ";
  20 INPUT N
  30 PRINT N
  40 PRINT "TOMA MAXIMA ";
  50 INPUT I
  60 PRINT I
  70 PRINT "QUIEN JUEGA 1ER : JR
(0);ZX (1)"
  75 INPUT D
 100 IF D THEN GO TO 200
 110 PRINT "SU TURNO ";
 120 INPUT P
 130 IF P<>INT P OR P<1 OR P>I O
R P>N THEN GO TO 110
 140 LET D=SGN PI
 150 GO TO 1000
 200 LET S=I+D
 210 LET T=(N-S)/(I+D)
 220 IF T=INT T THEN GO TO 250
 230 LET S=S-D
 240 GO TO 210
 250 LET P=ABS (S-D)
 260 IF NOT P THEN LET P=D
 270 LET D=NOT PI
1000 PRINT P
1010 LET N=N-P
1020 PRINT "QUEDAN ";N
1030 PAUSE 120
1040 IF NOT N THEN GO TO 2000
1050 GO TO 100
2000 CLS
2005 PRINT "BRAVO"
2010 IF D THEN PRINT "PARA MI"
```

```
TOTAL 13
TOMA MAXIMA 4
QUIEN JUEGA 1ER : JR (0);ZX (1)
2
QUEDAN 11
SU TURNO 4
QUEDAN 7
1
QUEDAN 6
SU TURNO 3
QUEDAN 3
2
QUEDAN 1
SU TURNO
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT PAPER 6;"TOTAL ";: BE
EP .2,10
  20 INPUT N
  30 PRINT N
  40 PRINT PAPER 4;"TOMA MAXIMA
";: BEEP .2,10
  50 INPUT I
  60 PRINT I
  70 PRINT PAPER 5;"QUIEN JUEGA
1ER : JR (0);ZX (1)": BEEP .2,10
  75 INPUT D
 100 IF D THEN GO TO 200
 110 PRINT INK 5; FLASH 1;"SU TU
RNO "
 120 INPUT P
 130 IF P<>INT P OR P<1 OR P>I O
R P>N THEN GO TO 110
 140 LET D=1
 150 GO TO 1000
 200 LET S=I+D
 210 LET T=(N-S)/(I+D)
 220 IF T=INT T THEN GO TO 250
 230 LET S=S-D
 240 GO TO 210
 250 LET P=ABS (S-D)
 260 IF NOT P THEN LET P=D
 270 LET D=0
1000 PRINT P
1010 LET N=N-P
1020 PRINT BRIGHT 1; PAPER 3;"QU
EDAN ";N
1030 PAUSE 120
1040 IF NOT N THEN GO TO 2000
1050 GO TO 100
2000 CLS
2005 PRINT PAPER 5;"BRAVO"
2010 IF D THEN PRINT PAPER 5;"PA
RA MI"
```

```
TOTAL 24
TOMA MAXIMA 4
QUIEN JUEGA 1ER : JR (0);ZX (1)
3
QUEDAN 21
SU TURNO
4
QUEDAN 17
1
QUEDAN 16
SU TURNO
3
QUEDAN 13
2
QUEDAN 11
SU TURNO
4
QUEDAN 7
1
QUEDAN 6
SU TURNO
```

Cerillas para varios

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: fácil.

Categoría del juego: lógica (para varios jugadores).

Dificultad del juego: bastante difícil.

EL JUEGO: En este juego para varios jugadores, el que dirige el juego deberá, al principio de la partida, escoger un número de cerillas e indicar al ZX el número de jugadores.

En cada jugada, el jugador que tiene el turno está autorizado a retirar del montón de cerillas que quedan hasta el doble de lo que acaba de retirar el jugador que le precede; el que retira la última cerilla gana (o pierde, si se modifican ligeramente las reglas).

EL PROGRAMA: Las líneas 1Ø a 5Ø permiten las diferentes entradas.

Las líneas 1ØØ a 13Ø efectúan las impresiones.

La línea 15Ø controla la jugada de cada jugador (en relación con la jugada precedente).

La línea 2Ø2 administra las jugadas (rebaja en 1 el número del jugador).

La línea 2ØØ realiza la comprobación del final, y la línea 22Ø visualiza el número del jugador que gana (con la regla prevista al inicio).

PARA JUGAR: Después de poner en marcha el programa con RUN, se introduce el número de cerillas inicial, seguido de NEWLINE, y después el número de jugadores (y NEWLINE).

El primer jugador podrá solamente tomar una cerilla; el segundo podrá quitar 1 o 2; el siguiente, 1, 2 o 1, 2, 3 o 4, según la jugada anterior, y así sucesivamente; el ZX visualiza cada vez el número máximo autorizado.

Cuando se retira la última cerilla, se visualiza EL JUGADOR ... HA GANADO, y la partida se termina.

POSIBLES EXTENSIONES: Puede decidirse que el que tome la última cerilla queda eliminado del juego y volver a empezar la partida con los jugadores restantes, los cuales se irán eliminando de esta manera, unos detrás de otros.

Puede considerarse una representación gráfica de montones de cerillas, decidir al azar el número a retirar al inicio de la partida, etc.

CERILLAS PARA VARIOS

PROGRAMA ZX 81

```
  10 PRINT "NUMERO DE CERILLAS "
;
  20 INPUT N
  30 PRINT N
  40 PRINT "NUMERO DE JUGADORES
"
  50 INPUT J
  60 LET F=SGN PI
  70 LET K=J
  80 CLS
 100 PRINT AT 4,SGN PI;"JUGADOR
NO. ";K
 110 PRINT AT 6,SGN PI;"NUM. MAX
 A SACAR ";F
 120 PRINT AT 8,SGN PI;"NUM. TOT
AL ";N
 130 PRINT AT 12,SGN PI;"SU TURN
O ";
 140 INPUT C
 150 IF C=NOT PI OR C>F OR C>N T
HEN GO TO 130
 180 LET N=N-C
 190 LET F=C+C
 200 IF NOT N THEN GO TO 220
 202 IF NOT (K-SGN PI) THEN LET
K=J
 205 LET K=K-SGN PI
 210 GO TO 80
 220 PRINT AT 12,SGN PI;"EL JUGA
DOR ";K;" HA GANADO"
```

```
JUGADOR NO. 3

NUM. MAX A SACAR 10

NUM. TOTAL 28


SU TURNO
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "NUMERO DE CERILLAS "
;
  20 INPUT N
  30 PRINT N
  40 PRINT "NUMERO DE JUGADORES
"
  50 INPUT J
  60 LET F=1
  70 LET K=J
  80 CLS
 100 PRINT AT 4,1;"JUGADOR NO. "
;K
 110 PRINT AT 6,1;"NUM. MAX A SA
CAR ";F
 120 PRINT AT 8,1;"NUM. TOTAL ";
N
 130 PRINT AT 12,1;"SU TURNO ";
 140 INPUT C
 150 IF C=0 OR C>F OR C>N THEN G
O TO 130
 180 LET N=N-C
 190 LET F=C*2
 200 IF NOT N THEN GO TO 220
 202 IF NOT (K-1) THEN LET K=J
 205 LET K=K-1
 210 GO TO 80
 220 PRINT AT 12,1;"EL JUGADOR "
;K;" HA GANADO": BEEP .8,10
```

```
JUGADOR NO. 1

NUM. MAX A SACAR 10

NUM. TOTAL 4


SU TURNO
```

Calendario perpetuo

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: mediana.
Categoría del juego: reclamo publicitario.
Dificultad del juego: ninguna.

EL JUEGO: Este programa permitirá confeccionar todos los años su propio calendario; podrá utilizarse para enviar a los amigos un calendario del mes de su nacimiento o para felicitar el santo, etc.

EL PROGRAMA: Las líneas 2Ø y 4Ø permiten introducir los meses y el año.

Las líneas 5Ø y 6Ø calculan el número del primer día del mes.

La línea 7Ø calcula el número de días del mes (para simplificar, no se tiene en cuenta el 29 de febrero).

El bucle A (9Ø-11Ø) efectúa la impresión del calendario propiamente dicho.

PARA JUGAR: Después de poner en marcha el programa con la instrucción RUN, deberá introducirse el año bajo la forma, por ejemplo, 1983, seguido de NEWLINE; se obtiene entonces la pregunta MES, a la que es conveniente contestar con el número de mes siguiendo el orden normal (1 para ENERO, 2 para FEBRERO, etc.).

Se teclea a continuación NEWLINE, y el calendario se visualiza entonces en la pantalla.

Para guardar una copia, si se dispone, por supuesto, de una impresora, bastará con pulsar COPY al final de la visualización.

POSIBLES EXTENSIONES: Podrá preverse sin dificultad la visualización de los 29 de febrero, una opción que permita la visualización de un año completo, la introducción del nombre de los meses, etc.

CALENDARIO PERPETUO

PROGRAMA ZX 81

```
  10 PRINT "        ANO ";
  20 INPUT A
  25 PRINT A;
  30 PRINT "        MES ";
  40 INPUT M
  45 PRINT M
  50 LET D=INT (365.25*(A-(M<INT
 PI)))+INT (30.6*(M+SGN PI+(M<IN
T PI)*12))-INT ((INT ((A-(M<INT
PI))/100)-7)*.75)+SGN PI
  60 LET D=D-7*INT (D/7)
  70 LET M=31-(M=4 OR M=6 OR M=9
 OR M=11)-(M=2)*INT PI
  75 PRINT
  80 PRINT "DOM LUN MAR MIE JUE
VIE SAB    "
  85 PRINT
  90 FOR A=SGN PI TO M
 100 PRINT TAB 4*(D+A*SGN PI);A;
 105 IF D+A-SGN PI=6 THEN LET D=
D-7
 110 NEXT A
```

```
      ANO 1984        MES 12
```

DOM	LUN	MAR	MIE	JUE	VIE	SAB
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

PROGRAMA ZX SPECTRUM

```
  10 PRINT "        ANO ";
  20 INPUT A
  25 PRINT A;
  30 PRINT "        MES ";
  40 INPUT M
  45 PRINT M
  50 LET D=INT (365.25*(A-(M<3))
)+INT (30.6*(M+1+(M<3)*12))-INT
((INT ((A-(M<3))/100)-7)*.75)+1
  60 LET D=D-7*INT (D/7)
  70 LET M=31-(M=4 OR M=6 OR M=9
 OR M=11)-(M=2)*3
  75 PRINT : PRINT : PRINT
  80 PRINT INK 2; INVERSE 1;" D
OM LUN MAR MIE JUE VIE SAB   "
  85 PRINT
  90 FOR A=SGN PI TO M
 100 PRINT PAPER 6; INK 2; BRIGH
T 1;TAB 4*(D+A-1)+2;A;
 105 IF D+A-1=6 THEN LET D=D-7
 110 NEXT A
```

```
      ANO 1984        MES 5
```

		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

La ruleta

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: mediana.

Categoría del juego: de azar (para varios jugadores).

Dificultad del juego: fácil.

EL JUEGO: Es sobre todo espectacular por su aspecto gráfico: una bola, representada por un cuadrado negro, se desplaza en un terreno de juego que consta de casillas numeradas del 1 al 36, además de tres casillas suplementarias, el ØØ, una casilla P (para par) y una casilla I (para impar).

La bola recorre como mínimo una vez cada casilla, y después se para a la segunda vuelta; a continuación se vuelve a lanzar para otra jugada.

EL PROGRAMA: El bucle I (1Ø-3Ø) dibuja el terreno de juego, al que la línea 4Ø añade los tres símbolos P, ØØ e I.

El bucle I (11Ø-13Ø) realiza el desplazamiento de la bola; la parada se controla con las líneas 125 y 128.

La línea 135 efectúa un borrado particular si la bola pasa por los símbolos P, ØØ e I.

La línea 14Ø hace descender una línea.

La línea 16Ø comprueba la llegada al final del terreno de juego.

La línea 17Ø obtiene la posición de parada.

PARA JUGAR: Es suficiente con ejecutar RUN y mirar.

Una vez que la bola se para, basta con pulsar cualquier tecla (excepto BREAK) para volver a lanzarla. Para salir del programa se pulsa BREAK.

POSIBLES EXTENSIONES: Para que se parezca más a una ruleta, pueden intentarse diferentes acciones: dibujar mejor el terreno de juego, introducir colores, simular que la bola va más despacio, introducir distintas posibilidades de apuestas y asegurar la gestión de las ganancias correspondientes, llevar la contabilidad de varios jugadores y anunciar los resultados de una forma más usual.

LA RULETA

PROGRAMA ZX 81

```
   5 LET A=INT PI
   7 LET B=A-SGN PI
  10 FOR I=SGN PI TO A*A
  20 PRINT AT A,A*I+A;B*I;AT A+A
,A*I+A;B*I+18;AT A*A,A*I+A;B*I-S
GN PI;AT 12,A*I+A;B*I+17
  30 NEXT I
  32 LET X=NOT PI
  35 LET L=A+SGN PI
  40 PRINT AT L,SGN PI;"P";AT L+
A,SGN PI;"00";AT 10,SGN PI;"I"
 110 FOR I=SGN PI TO 10
 120 PRINT AT L,A*I;"    █"
 122 PAUSE CODE "="
 125 LET X=X-SGN PI
 128 IF NOT X THEN PAUSE 4E4
 130 NEXT I
 135 PRINT AT 5,SGN PI;" ";AT 8,
SGN PI;" ";AT 11,SGN PI;" ";AT 1
4,SGN PI;" "
 140 LET L=L+A
 160 IF L<>CODE "(" THEN GO TO 1
10
 170 LET X=INT (RND*CODE "PI")
 180 GO TO CODE "7"
```

```
        2   4   6   8   10  12  14  16  18
P

        20  22  24  26  28  30  32  34  36
00                █

        1   3   5   7   9   11  13  15  17
I

        19  21  23  25  27  29  31  33  35
```

PROGRAMA ZX SPECTRUM

```
   5 BORDER 3: LET A=3
   7 LET B=A-1
  10 FOR I=1 TO 9
  20 PRINT PAPER 2;AT A,A*I+A;B*
I; PAPER 4;AT A+A,A*I+A;B*I+18;
PAPER 2;AT 9,A*I+A;B*I-1; PAPER
4;AT 12,A*I+A;B*I+17
  30 NEXT I
  32 LET X=0
  35 LET L=A+1
  40 PRINT AT L,1;"P";AT L+A,1;"
00";AT 10,1;"I"
 110 FOR I=1 TO 10
 120 PRINT INK 5;AT L,A*I;"    █"
 122 BEEP 0.03,0: PAUSE 10
 125 LET X=X-1
 128 IF NOT X THEN BEEP 0.2,10:
BEEP 0.2,10: BEEP 0.2,10: PAUSE
4E4
 130 NEXT I
 135 PRINT AT 5,1;" ";AT 8,1;" "
;AT 11,1;" ";AT 14,1;" "
 140 LET L=L+A
 160 IF L<>18 THEN GO TO 110
 170 LET X=INT (RND*66)
 180 GO TO 35
```

```
        2   4   6   8   10  12  14  16  18
P

        20  22  24  26  28  30  32  34  36
00                    █

        1   3   5   7   9   11  13  15  17
I

        19  21  23  25  27  29  31  33  35
```

Batalla naval

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: fácil.
Categoría del juego: de reflexión.
Dificultad del juego: fácil.

EL JUEGO: Una batalla naval, en principio bastante simple, con un solo barco que se esconde al azar en una rejilla de 10 × 10 casillas.

En cada jugada se dispara dando las coordenadas H (horizontal) y V (vertical) de la casilla apuntada, y el ZX responde visualizando la distancia que separa el barco invisible de la casilla apuntada. Pero, si se dispara demasiado cerca del barco, existe el riesgo de que éste vuelva a esconderse al azar.

EL PROGRAMA: Los bucles I (2Ø-6Ø) y J (3Ø-5Ø) dibujan el terreno de juego.

Las líneas 7Ø y 8Ø definen el emplazamiento del barco.

Las coordenadas del disparo se introducen en las líneas 1ØØ y 12Ø, y la línea 13Ø visualiza la jugada en la pantalla. La distancia se calcula en la línea 14Ø, y la línea 15Ø comprueba si se ha dado en el blanco o no.

La línea 16Ø interviene para mover el barco, si los disparos se han acercado demasiado.

La línea 17Ø permite la visualización de la distancia.

PARA JUGAR: El terreno de juego se visualiza después de pulsar RUN. Cuando se visualiza H, se responde dando el número de la línea horizontal sobre la que se desea disparar. Se pulsa en seguida NEWLINE, y a la pregunta V se contesta dando el número de la columna vertical seguido de NEWLINE. Si se ha disparado demasiado cerca, se obtiene la visualización PFTT..., lo que significa que el barco enemigo ha cambiado de lugar.

Si se dispara sobre la casilla exacta, se obtiene la visualización del número total de disparos efectuados para ganar.

POSIBLES EXTENSIONES: Son innumerables, ya que la batalla naval es uno de los juegos más difundidos que existen. Pueden introducirse dos jugadores, jugar contra el ZX, poner varios barcos, submarinos, etc.

BATALLA NAVAL

PROGRAMA ZX 81

```
  10 LET A=NOT PI
  20 FOR I=A TO INT (PI*PI)
  30 FOR J=A TO INT (PI*PI)
  40 PRINT AT 2,J+J+INT PI;J;AT
INT PI+I,A;I;AT INT PI+I,J+J+INT
 PI;"."
  50 NEXT J
  60 NEXT I
  65 LET S=SGN PI
  70 LET X=INT (RND*10)
  80 LET Y=INT (RND*10)
  90 PRINT AT CODE "?",A;"H"
 100 INPUT H
 110 PRINT AT CODE "(",A;"V"
 120 INPUT V
 130 PRINT AT H+INT PI,V+V+INT P
I;S
 140 LET D=INT SQR ((X-H)*(X-H)+
(Y-V)*(Y-V))
 150 IF NOT D THEN GO TO CODE "C
OS "
 155 LET S=S+SGN PI
 157 IF D=SGN PI THEN PRINT AT C
ODE ")",A;"PFTT..."
 160 IF D=SGN PI THEN GO TO 70
 170 PRINT AT CODE ")",A;"DISTAN
CIA :";D
 180 GO TO 90
 200 CLS
 210 PRINT S;" DISPAROS"
```

```
   0 1 2 3 4 5 6 7 8 9
0  . . . . . . . . . .
1  . . . . . . . . . .
2  . . . . 1 . . . . .
3  . . . . . . . . . .
4  . . . . . . . . . .
5  . . . . . . . . . .
6  . . . . . . . . . .
7  . . . . . . . . . .
8  . . . . . . . . . .
9  . . . . . . 2 . . .



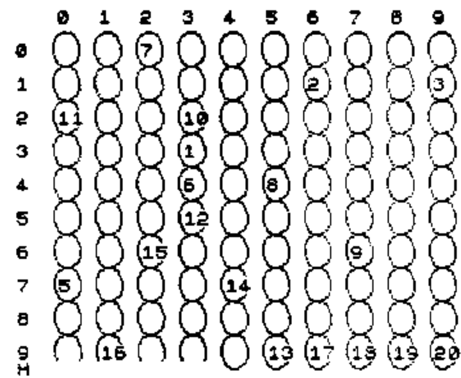
H
V
DISTANCIA :8
```

PROGRAMA ZX SPECTRUM

```
  10 LET A=0
  20 FOR I=A TO 9
  30 FOR J=A TO 9
  40 PRINT AT 0,3*J+3;J;AT 2+2*I
,A;I: CIRCLE INK 2;5+256*(3*J+3)
/32,180-(176*(3+2*I)/22),8
  50 NEXT J
  60 NEXT I
  65 LET S=1
  70 LET X=INT (RND*10)
  80 LET Y=INT (RND*10)
  90 PRINT AT 21,A;"H
 "

 100 INPUT H
 110 PRINT AT 21,A;"V
   "

 120 INPUT V
 130 PRINT AT 2*H+2,3*V+3;S
 140 LET D=INT SQR ((X-H)*(X-H)+
(Y-V)*(Y-V))
 150 IF NOT D THEN GO TO 200
 155 LET S=S+1
 157 IF D=1 THEN PRINT AT 21,A;"
PFTT...            ": PAUSE 20
 160 IF D=1 THEN GO TO 70
 170 PRINT AT 21,A;"DISTANCIA :"
;D: BEEP 0.5,10: PAUSE 150
 180 GO TO 90
 200 CLS
 210 PRINT S;" DISPAROS"
```



Golf

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: mediana.
Categoría del juego: solitario.
Dificultad del juego: bastante fácil.

EL JUEGO: Se trata de realizar con el mínimo de golpes posibles un recorrido de golf de 18 agujeros. Para realizarlo, es necesario golpear la pelota con más o menos fuerza hasta que caiga en el agujero.

Pero, ¡cuidado!, puede haber un cierto riesgo en cada golpe, y a menudo se tendrá que volver hacia atrás (entonces deberá colocarse un signo – antes de la indicación de la fuerza).

EL PROGRAMA: La línea 25 permite definir la posición del agujero, y la línea 3Ø dibuja el campo de juego.

En la línea 8Ø se calcula el desplazamiento que se realiza.

El bucle I (9Ø-12Ø) efectúa el movimiento de la pelota.

La línea 13Ø comprueba si la jugada es buena o no; si lo es, vuelve a empezarse (línea 22Ø) hasta que se hayan efectuado los 18 agujeros (bucle S (15-22Ø); fin del bucle en 24Ø, con visualización de fin).

PARA JUGAR: Se obtiene la visualización del campo pulsando RUN, así como el dibujo de la pelota en posición de salida y el texto: AGUJERO 1. FUERZA; se responderá entonces dando la fuerza que se desee (un número entre 1 y 3ØØ...).

Después de pulsar NEWLINE, la pelota empieza a moverse; puede pararse delante del agujero (en este caso se vuelve a empezar), caer dentro (se pasa al siguiente agujero) o pasarse: se golpeará entonces con una fuerza negativa.

El juego se termina con la visualización 18 AGUJEROS EN ... GOLPES.

POSIBLES EXTENSIONES: Es posible ampliar el juego para que participen varios jugadores, mejorar la calidad gráfica, utilizar una representación espacial del campo (remitirse a determinados juegos de arcada), tener un campo más accidentado, obstáculos, cambiar de club, etc.

GOLF

PROGRAMA ZX 81

```
  10 LET E=NOT PI
  15 FOR S=SGN PI TO 18
  20 LET X=SGN PI
  25 LET T=INT (21*RND+9)
  30 PRINT AT 21,SGN PI;"[redacted]";AT 21,T;" ";AT 20,X;"O"
  35 PRINT AT 5,SGN PI;"AGUJERO ";S;"  FUERZA  "
  60 INPUT F
  70 PRINT AT 5,19;F
  80 LET F=INT (F*.16-((ABS F)*(SQR (RND)+.1)**2)/1000)
  85 IF NOT F THEN GO TO 35
  90 FOR I=SGN F TO F STEP SGN F
 100 LET X=X+SGN F
 110 IF X<31 AND X>=SGN PI THEN PRINT AT 20,X-SGN F;" ";AT 20,X;"O"
 120 NEXT I
 130 IF X=T THEN GO TO 200
 140 LET E=E+SGN PI
 150 GO TO 35
 200 PRINT AT 20,X;" ";AT 21,X;"O"
 210 PAUSE 200
 220 NEXT S
 230 CLS
 240 PRINT "18 AGUJEROS EN ";E;" GOLPES"


AGUJERO 1  FUERZA 40
```

PROGRAMA ZX SPECTRUM

```
  10 LET E=0
  15 FOR S=1 TO 18
  20 LET X=1
  25 LET T=INT (21*RND+9)
  30 PRINT INK 3;AT 21,1;"[redacted]"; INK 0;AT 21,T;" ";AT 20,X;"O"
  35 PRINT AT 5,1;"AGUJERO ";S;" FUERZA"
  60 INPUT F
  70 PRINT AT 5,19;F
  80 LET F=INT (F*.16-((ABS F)*(SQR (RND)+.1)↑2)/1000)
  85 IF NOT F THEN GO TO 35
  90 FOR I=SGN F TO F STEP SGN F
 100 LET X=X+SGN F: PAUSE 15
 110 IF X<31 AND X>=1 THEN PRINT AT 20,X-SGN F;" ";AT 20,X;"O": BEEP 0.03,0
 120 NEXT I
 130 IF X=T THEN GO TO 200
 140 LET E=E+1
 150 GO TO 35
 200 BEEP 1,10: PRINT AT 20,X;" ";AT 21,X;"O"
 210 PAUSE 200
 220 NEXT S
 230 CLS
 240 PRINT "18 AGUJEROS EN ";E;" GOLPES"


AGUJERO 4  FUERZA 120
```

Solfeo

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: de reflexión.
Dificultad del juego: fácil.

EL JUEGO: Este programa propone una iniciación elemental al solfeo, permitiendo aprender a reconocer fácilmente las diferentes notas, que aparecen unas detrás de otras en un pentagrama; después de la visualización de la nota, debe introducirse su nombre y se visualizará el marcador permanentemente.

El programa está previsto inicialmente para la clave de sol, pero puede adaptarse muy fácilmente a cualquier otra clave.

EL PROGRAMA: El bucle I (15-3Ø) dibuja el pentagrama.

La línea 4Ø escoge la nota y se visualiza en la línea 5Ø.

La línea 6Ø visualiza el marcador.

La línea 7Ø visualiza el nombre de la clave (aquí es la clave de sol) y realiza la pregunta.

La línea 85 se ocupa del fin de la partida.

Las líneas 9Ø y 1ØØ descifran la nota introducida.

La línea 12Ø controla el resultado.

PARA JUGAR: Después de haber puesto en marcha el programa con RUN, se visualizará el pentagrama, así como la primera nota. Se contestará a la pregunta ¿NOTA? dando el nombre de la nota (do, re, mi, fa, sol, la o si), seguido de NEWLINE.

El marcador indicará si es correcto o no y se visualizará una nueva nota. Para terminar, bastará con responder Ø seguido de NEWLINE a la pregunta ¿NOTA?

POSIBLES EXTENSIONES: Puede cambiarse fácilmente el programa para trabajar con distintas claves.

También puede mejorarse el grafismo, introducir color, el sonido (por ejemplo, si se contesta correctamente, se escuchará la nota).

También podrá realizarse, siguiendo el mismo principio, un juego que permita reconocer acordes, etc.

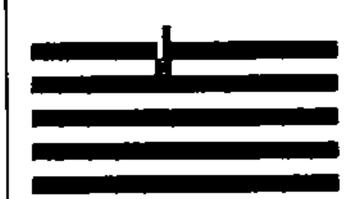
SOLFEO

PROGRAMA ZX 81

```
   5 LET S=NOT PI
   7 LET P=S
  10 CLS
  15 FOR I=SGN PI TO 5
  20 PRINT AT I+I,NOT PI;"██████████████████████"
  30 NEXT I
  40 LET X=INT (12*RND)
  50 PRINT AT X-SGN PI,9;"▌";AT
X,9;"▌";AT X+SGN PI,9;"█"
  60 PRINT AT 13,NOT PI;"MARCADO
R ";S;" SOBRE ";P;" INTENTOS"
  70 PRINT AT 15,NOT PI;"CLAVE D
E SOL; NOTA ?"
  80 INPUT R$
  85 IF R$="0" THEN STOP
  90 LET Y=(SGN PI AND R$="FA")+
(2 AND R$="MI")+(INT PI AND R$="
RE")
 100 LET Y=Y+(4 AND R$="DO")+(5
AND R$="SI")+(6 AND R$="LA")
 110 LET P=P+SGN PI
 120 IF Y=X OR Y=X-7 THEN LET S=
S+SGN PI
 130 GOTO 10
```

PROGRAMA ZX SPECTRUM

```
   5 LET S=0
   7 LET P=S
  10 PAPER 6: CLS
  15 FOR I=1 TO 5
  20 PRINT INK 5;AT I+I,0;"██████████████████████"
  30 NEXT I
  40 LET X=INT (12*RND)
  50 PRINT INK 2; INVERSE 1;AT X
-1,9;"▌";AT X,9;"▌";AT X+1,9;"O"
  60 PRINT AT 13,0;"MARCADOR :";
S;" SOBRE ";P;" INTENTOS"
  70 PRINT AT 15,0;"CLAVE DE SOL
; NOTA :";
  80 INPUT R$: PRINT R$
  85 IF R$="0" THEN STOP
  90 LET Y=(1 AND R$="FA")+(2 AN
D R$="MI")+(3 AND R$="RE")
 100 LET Y=Y+(4 AND R$="DO")+(5
AND R$="SI")+(6 AND R$="LA")
 110 LET P=P+1
 120 IF Y=X OR Y=X-7 THEN LET S=
S+1: BEEP 1,20-4*X: PAUSE 100
 130 GO TO 10
```



```
MARCADOR :5 SOBRE 5 INTENTOS

CLAVE DE SOL; NOTA :MI
```

El juego del 21

REFERENCIAS:

Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: contra el ZX.
Dificultad del juego: fácil.

EL JUEGO:

Deberá introducirse una apuesta al principio del juego, y podrán tirarse los dados tecleando 1; en cada tirada se obtiene un nuevo total que aumentará poco a poco acercándose a 21. Se intentará aproximarse a este número lo máximo posible, sin pasar de él. Cuando se decida parar, bastará con teclear Ø y se dejará jugar al ZX, que intentará obtener entonces una mejor puntuación.

Al final de cada partida se visualizán las ganancias, así como su total (o de pérdidas).

EL PROGRAMA:

Las líneas 6Ø, 7Ø y 8Ø realizan el control de quién debe jugar (jugador o el ZX).

El subprograma (1ØØØ-1Ø1Ø) tira los dados.

Las líneas 11Ø y 12Ø calculan el total obtenido por el jugador y lo visualizan.

El total del ZX se calcula en la línea 17Ø y se visualiza en la línea 175.

La línea 18Ø controla el final del juego para el ZX.

Las líneas 2ØØ a 21Ø visualizan los resultados, y la línea 22Ø busca quién es el ganador.

La ganancia (o la pérdida) se visualizará con la línea 26Ø, y el total con la línea 5ØØ.

La línea 52Ø controla el final de la partida.

PARA JUGAR:

Pulsar RUN para poner en marcha el juego.

Ante la pregunta SU APUESTA, introducir la apuesta inicial, seguida de NEWLINE, y teclear 1 para lanzar el dado. La primera tirada se visualiza bajo la forma TIENE ..., y se volverá a teclear 1 seguido de NEWLINE tantas veces como se quiera volver a tirar.

Cuando se termina, se teclea Ø y NEWLINE para dejar jugar al ZX (que hace la función de la banca). La ganancia (o la pérdida, si está con un signo –) se visualizará al final del juego, así como el total de ganancias (o pérdidas) obtenidas desde el principio.

Si se desea parar el juego, se tecleará N como respuesta a la pregunta SIGUE ¿(S o N)?

POSIBLES EXTENSIONES: Evidentemente, este juego será mucho más interesante si lo practican varios jugadores; por consiguiente, será fácil modificar el programa; también puede dibujarse el dado o, mejor aún, utilizar cartas, lo que permitirá jugar al BLACK JACK.

EL JUEGO DEL 21

PROGRAMA ZX 81

```
  10 LET G=NOT PI
  20 CLS
  30 PRINT "SU APUESTA ";
  40 INPUT M
  50 PRINT M
  55 LET T=G-G
  60 PRINT AT 5,1;"1 :TIRAR ; 0
TERMINAR"
  70 INPUT R
  80 IF NOT R THEN GO TO 130
 100 GO SUB 1000
 110 LET T=T+X
 120 PRINT AT 9,1;"TOTAL :";T
 125 GO TO 70
 130 CLS
 140 PRINT "LA BANCA JUEGA"
 150 LET S=R-R
 160 GO SUB 1000
 170 LET S=S+X
 175 PRINT AT 5,1;"TOTAL :";S
 177 PAUSE 120
 180 IF S>T OR S>=CODE ")" THEN
GO TO CODE "CODE "
 190 GO TO 160
 196 CLS
 200 PRINT AT 9,1;"TIENE ";T
 210 PRINT AT 11,1;"LA BANCA TIE
NE ";S
 220 IF S>CODE "+" OR (S<T AND T
<=CODE "+") THEN GO TO 260
 240 LET M=-M
 260 PRINT AT 14,1;"CANANCIA ";M
 270 LET G=G+M
 500 PRINT AT 18,1;"TOTAL ";G
 505 PRINT "SIGUE (S O N)?"
 510 INPUT R$
 520 IF R$="N" THEN STOP
 530 GO TO CODE "="
1000 LET X=INT (RND*6+1)
1010 RETURN
```

```
SU APUESTA 10


 1 :TIRAR ; 0 :TERMINAR


 TOTAL :11

 TIENE 11

 LA BANCA TIENE 15

 CANANCIA -10


 TOTAL -10
SIGUE (S O N)?
```

PROGRAMA ZX SPECTRUM

```
  10 LET G=0
  20 CLS
  30 PRINT "SU APUESTA ";
  40 INPUT M
  50 PRINT M
  55 LET T=0
  60 PRINT AT 5,1;"1 :TIRAR ; 0
:TERMINAR"
  70 INPUT R
  80 IF NOT R THEN GO TO 130
 100 GO SUB 1000
 110 LET T=T+X
 120 PRINT AT 9,1;"TOTAL :";T
 125 GO TO 70
 130 CLS
 140 PRINT "LA BANCA JUEGA"
 150 LET S=0
 160 GO SUB 1000
 170 LET S=S+X
 175 PRINT AT 5,1;"TOTAL :";S
 177 PAUSE 120
 180 IF S>T OR S>=17 THEN GO TO
196
 190 GO TO 160
 196 CLS
 200 PRINT AT 9,1;"TIENE ";T: BE
EP 0.05,0
 210 PRINT AT 11,1;"LA BANCA TIE
NE ";S: BEEP 0.05,0
 220 IF S>21 OR (S<T AND T<=21)
THEN GO TO 260
 240 LET M=-M
 260 PRINT AT 14,1;"CANANCIA ";M
 270 LET G=G+M
 500 PRINT AT 18,1;"TOTAL ";G
 505 PRINT "SIGUE (S O N)?"
 510 INPUT R$
 520 IF R$="N" THEN STOP
 530 GO TO 20
1000 LET X=INT (RND*6+1): BEEP 1
,7*X-22
1010 RETURN
```

```
 TIENE 19

 LA BANCA TIENE 17

 CANANCIA 100


 TOTAL 100
SIGUE (S O N)?
```

Par-impar

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: fácil.
Categoría del juego: tirada de dados.
Dificultad del juego: fácil.

EL JUEGO: Es un juego de azar para uno o varios jugadores. El ZX tira simultáneamente dos dados que no son visibles y hace tres preguntas: cuál es la paridad del total, cuál es el total y, por último, dar el valor de uno de los dados; luego puede continuarse el juego o interrumpirlo en cualquier momento.

EL PROGRAMA: Los dados se tiran por un subprograma que se encuentra en las líneas 1ØØØ y 1Ø1Ø.

La línea 5Ø anota el número de partidas, y las líneas 15Ø, 21Ø y 25Ø contabilizan los puntos marcados.

La paridad del total se determina por las líneas 97 y 1ØØ (la línea 1ØØ compara con la respuesta del jugador).

La línea 4ØØ efectúa la visualización del final de la partida: total, número de puntos, número de partidas jugadas.

La línea 42Ø provoca la parada si se responde N a la pregunta SE CONTINÚA.

PARA JUGAR: Se pone en marcha el programa con la instrucción RUN.

Basta después con responder a las preguntas hechas, siguiendo escrupulosamente las indicaciones dadas y pulsando NEWLINE después de cada respuesta.

Cuando una partida se termina, se obtiene la visualización del total actual y del número de partidas bajo la forma de, por ejemplo, 6/1Ø.

Entonces se obtiene la visualización de la pregunta SE CONTINÚA (S o N), y se responderá N seguido de NEWLINE para terminar.

POSIBLES EXTENSIONES: Si quieren jugar varias personas, será interesante prever la posibilidad de apostar y de añadir una gestión de la puntuación realizada por cada jugador.

También se podrá prever una mejora en el plan de la presentación (utilización del color, etc.).

PAR-IMPAR

PROGRAMA ZX 81

```
  10 LET T=NOT PI
  15 LET R=T
  20 GO SUB 1000
  30 LET A=X
  40 GO SUB 1000
  50 LET R=R+SGN PI
  60 LET S=A+X
  70 CLS
  80 PRINT "TOTAL PAR (0) O IMPA
R (1) :";
  90 INPUT N
  95 PRINT N
  97 LET P=S/2-INT (S/2)
 100 IF N=P OR N-.5=P THEN GO TO
 140
 110 PRINT AT 5,1;"FALSO"
 120 GO TO 160
 140 PRINT AT 5,1;"EXACTO"
 150 LET T=T+SGN PI
 160 PRINT AT 8,1;"SU TOTAL :";
 170 INPUT N
 175 PRINT N
 180 IF N=S THEN GO TO 210
 190 PRINT AT 12,1;"HA PERDIDO"
 195 PAUSE 130
 200 GO TO 400
 210 LET T=T+2
 220 PRINT AT 12,1;"BRAVO; DAR U
N DADO ?"
 230 INPUT N
 240 IF N<>A AND N<>S-A THEN GO
TO 190
 250 LET T=T+INT PI
 260 GO TO 195
 400 CLS
 405 PRINT "TOTAL :";T;"/";R;" ;
CONTINUA (S O N)"
 410 INPUT R$
 420 IF R$="N" THEN STOP
 430 GO TO 20
1000 LET X=INT (RND*6+1)
1010 RETURN
```

```
TOTAL PAR (0) O IMPAR (1) :1


 FALSO

 SU TOTAL :8


 BRAVO; DAR UN DADO ?
```

PROGRAMA ZX SPECTRUM

```
  10 LET T=0
  15 LET R=T
  20 GO SUB 1000
  30 LET A=X
  40 GO SUB 1000
  50 LET R=R+1
  60 LET S=A+X: BEEP 1,3*S-20
  70 CLS
  80 PRINT "TOTAL PAR (0) O IMPA
R (1) :";
  90 INPUT N
  95 PRINT N
  97 LET P=S/2-INT (S/2)
 100 IF N=P OR N-.5=P THEN GO TO
 140
 110 PRINT AT 5,1;"FALSO"
 120 GO TO 160
 140 PRINT AT 5,1;"EXACTO"
 150 LET T=T+1
 160 PRINT AT 8,1;"SU TOTAL :";
 170 INPUT N
 175 PRINT N
 180 IF N=S THEN GO TO 210
 190 PRINT AT 12,1;"HA PERDIDO"
 195 PAUSE 130
 200 GO TO 400
 210 LET T=T+2
 220 PRINT AT 12,1;"BRAVO; DAR U
N DADO ?"
 230 INPUT N
 240 IF N<>A AND N<>S-A THEN GO
TO 190
 250 LET T=T+3: GO TO 195
 400 CLS
 405 PRINT "TOTAL :";T;"/";R;" ;
CONTINUA (S O N)"
 410 INPUT R$
 420 IF R$="N" THEN STOP
 430 GO TO 20
1000 LET X=INT (RND*6+1)
1010 RETURN
```

```
TOTAL PAR (0) O IMPAR (1) :0


 EXACTO

 SU TOTAL :7


 HA PERDIDO
```

Simón

REFERENCIAS:

Modalidad: FAST o SLOW.

Dificultad de programación: fácil.

Categoría del juego: juego de memoria (uno o más jugadores).

Dificultad del juego: mediana.

EL JUEGO:

Aparecerá una serie de números, uno a uno.

El jugador deberá, en cada jugada, repetirlos todos desde el principio (lo que significa que tendrá que teclear uno más cada vez).

Intentará realizarlo el máximo de veces posible. Después de cada partida se guarda en memoria la mejor puntuación, lo que permitirá, llegado el caso, que jueguen varias personas.

EL PROGRAMA:

La línea 15 llama a un subprograma (líneas 9ØØ a 1Ø1Ø) que marca las reglas del juego.

La línea 2Ø confecciona las cifras, que a continuación son almacenadas en una variable alfanumérica A$ (línea 3Ø).

La línea 4Ø permite la impresión de la última cifra al azar en la sexta línea de la pantalla, y la duración de esta visualización está determinada por la línea 5Ø.

Las líneas 8Ø y 9Ø esperan la respuesta del jugador, la cual se compara con la respuesta correcta en la línea 1ØØ.

La línea 14Ø marca los puntos.

La línea 15Ø controla la mejor puntuación.

Las visualizaciones del final de la partida se realizan desde la línea 2ØØ a la línea 25Ø.

PARA JUGAR:

La puesta en marcha con RUN provoca la visualización de las reglas del juego, y después la de una cifra que desaparece. Entonces el aparato realiza la pregunta ¿SU SECUENCIA?

Se contesta tecleando la cifra, y a continuación NEWLINE. Si es correcta, aparece un instante la visualización de EXACTO. ATENCIÓN, y se muestra una nueva cifra, que se borra, y de nuevo la pregunta ¿SU SECUENCIA? Entonces se vuelven a teclear las dos cifras ya obtenidas, seguidas de NEWLINE. Se continúa así (3 cifras, después 4, 5, etc.), hasta que se comete un fallo, lo que provoca la visualización de la puntuación (y de la mejor puntuación obtenida hasta este momento), seguida de la pregunta OTRA PARTIDA ¿(S o N)?; para continuar se teclea S.

Puede salirse del programa en cualquier momento pulsando BREAK antes de la visualización de una cifra o tecleando una letra, seguida de NEWLINE.

POSIBLES EXTENSIONES: Pueden realizarse juegos parecidos, más o menos perfeccionados, sustituyendo las cifras por letras, por colores, por sonidos, etcétera.

SIMÓN

PROGRAMA ZX 81

```
  10 LET M=NOT PI
  12 LET A$=""
  15 GO SUB 900
  17 LET H=NOT PI
  18 PAUSE 90
  20 LET X=INT (RND*10
  30 LET A$=A$+STR$ X
  40 PRINT AT 6,RND*33,X
  50 PAUSE 20+20*RND
  70 GO SUB 900
  80 PRINT AT 8,1;"SU SECUENCIA
?"
  90 INPUT R$
 100 IF R$<>A$ THEN GO TO 200
 120 PRINT AT 10,1;"EXACTO . ATE
NCION"
 140 LET H=H+SGN PI
 150 IF M<H THEN LET M=H
 160 GO TO 18
 200 CLS
 210 PRINT "SU PUNTUACION ";H
 220 PRINT "MEJOR PUNTUACION ";M
 230 PRINT "OTRA PARTIDA (S O N)
?"
 240 INPUT R$
 250 IF R$="N" THEN STOP
 260 GO TO 12
 900 CLS
1000 PRINT AT 3,1;"DEBE REPETIR
TODOS LOS NUMEROS    (UNO MAS CAD
A VEZ)"
1010 RETURN
```

```
DEBE REPETIR TODOS LOS NUMEROS
 (UNO MAS CADA VEZ)


SU SECUENCIA ?

EXACTO . ATENCION
```

PROGRAMA ZX SPECTRUM

```
  10 LET M=0
  12 LET A$=""
  15 GO SUB 900
  17 LET H=0
  18 PAUSE 90
  20 LET X=INT (RND*10)
  30 LET A$=A$+STR$ X
  40 PRINT AT 6,RND*33;X: BEEP 1
,5*X-20
  50 PAUSE 20+20*RND
  70 GO SUB 900
  80 PRINT AT 8,1;"SU SECUENCIA
?"
  90 INPUT R$
 100 IF R$<>A$ THEN GO TO 200
 120 PRINT AT 10,1;"EXACTO . ATE
NCION"
 140 LET H=H+1
 150 IF M<H THEN LET M=H
 160 GO TO 18
 200 CLS
 210 PRINT "SU PUNTUACION ";H
 220 PRINT "MEJOR PUNTUACION ";M
 230 PRINT "OTRA PARTIDA (S O N)
?"
 240 INPUT R$
 250 IF R$="N" THEN STOP
 260 GO TO 12
 900 CLS
1000 PRINT AT 3,1;"DEBE REPETIR
TODOS LOS NUMEROS    (UNO MAS CAD
A VEZ)"
1010 RETURN
```

```
DEBE REPETIR TODOS LOS NUMEROS
 (UNO MAS CADA VEZ)

        3
```

Adivinar una cifra

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: fácil.
Categoría del juego: tirar los dados.
Dificultad del juego: fácil.

EL JUEGO: Es un juego de azar, de concepción muy simple y, sin embargo, apasionante. El jugador introduce su dinero y apuesta sobre una cifra del 1 al 6. Se tiran tres dados; si la cantidad apostada aparece una vez, el jugador gana su apuesta; si aparece dos veces, dobla su apuesta, y si aparece tres veces, triplica su apuesta.

EL PROGRAMA: El lanzamiento de los dados se realiza por un subprograma situado en las líneas 1ØØØ y 1Ø1Ø.

La visualización se efectúa por las líneas 15Ø y 16Ø.

Las líneas 18Ø a 21Ø calculan las ganancias en las distintas configuraciones posibles.

El total de las ganancias se efectúa en la línea 23Ø y el de las pérdidas en la línea 51Ø.

La variable R$, que se comprueba en la línea 62Ø, determina la posible continuación del juego.

PARA JUGAR: Se pone en marcha el programa con RUN, lo que provoca una primera visualización ¿SU APUESTA?; se contesta con un número seguido de NEWLINE. Se obtiene entonces una segunda visualización ¿UNA CIFRA DEL 1 AL 6?; se introduce una cifra seguida de NEWLINE. Los tres dados identificados por las letras A, B y C se visualizan, seguidos por un mensaje que indica la ganancia o la pérdida. A continuación aparece una indicación sobre el total actual, así como la pregunta SE CONTINÚA ¿(S o N)?; se tecleará S para continuar y N para parar (seguido de NEWLINE).

POSIBLES EXTENSIONES: Fácilmente podrán introducirse controles (apuesta mínima, tirada propuesta del 1 al 6, etc.). También podrán dibujarse los dados con un grafismo más elaborado, introducir la posibilidad de jugar varios, hacer participar igualmente al ZX, etc.

ADIVINAR UNA CIFRA

PROGRAMA ZX 81

```
   5 LET G=NOT PI
   7 CLS
  10 PRINT "SU APUESTA :";
  20 INPUT N
  30 PRINT N
  50 PRINT AT 5,1;"UNA CIFRA DEL
 1 AL 6 :";
  60 INPUT D
  80 PRINT D
  90 GO SUB 1000
 100 LET A=X
 110 GO SUB 1000
 120 LET B=X
 130 GO SUB 1000
 140 LET C=X
 150 PRINT AT 9,4;A;TAB 8;B;TAB
12;C
 160 PRINT AT 11,4;"A";TAB 8;"B"
;TAB 12;"C"
 170 LET T=D-D
 180 IF D=A THEN LET T=T+N
 190 IF D=B THEN LET T=T+N
 200 IF D=C THEN LET T=T+N
 210 IF T=0 THEN GO TO 500
 220 PRINT AT 15,1;"HA GANADO ";
T
 230 LET G=G+T
 240 GO TO 600
 500 PRINT AT 15,1;"HA PERDIDO"
 510 LET G=G-N
 600 PRINT AT 17,1;"TOTAL ACTUAL
:";G;"; CONTINUA (S O N)?"
 610 INPUT R$
 620 IF R$="N" THEN STOP
 630 GO TO 7
1000 LET X=INT (RND*6+1)
1010 RETURN
```

```
SU APUESTA :10


 UNA CIFRA DEL 1 AL 6 :3


    2    1    5

    A    B    C


 HA PERDIDO

 TOTAL ACTUAL :-10; CONTINUA (S
O N)?
```

PROGRAMA ZX SPECTRUM

```
   5 LET G=0
   7 CLS
  10 PRINT "SU APUESTA :";
  20 INPUT N
  30 PRINT N
  50 PRINT AT 5,1;"UNA CIFRA DEL
 1 AL 6 :";
  60 INPUT D
  80 PRINT D
  90 GO SUB 1000
 100 LET A=X
 110 GO SUB 1000
 120 LET B=X
 130 GO SUB 1000
 140 LET C=X
 150 PRINT INK 2;AT 9,4;A: BEEP
1,7*A-22: PRINT INK 4;AT 9,8;B:
BEEP 1,7*B-22: PRINT INK 5;AT 9,
12;C: BEEP 1,7*C-22
 160 PRINT FLASH 1;AT 11,4;"A";A
T 11,8;"B";AT 11,12;"C"
 170 LET T=0
 180 IF D=A THEN LET T=T+N
 190 IF D=B THEN LET T=T+N
 200 IF D=C THEN LET T=T+N
 210 IF T=0 THEN GO TO 500
 220 PRINT AT 15,1;"HA GANADO ";
T
 230 LET G=G+T
 240 GO TO 600
 500 PRINT AT 15,1;"HA PERDIDO"
 510 LET G=G-N
 600 PRINT AT 17,1;"TOTAL ACTUAL
:";G;"; CONTINUA (S O N)?"
 610 INPUT R$
 620 IF R$="N" THEN STOP
 630 GO TO 7
1000 LET X=INT (RND*6+1)
1010 RETURN
```

```
SU APUESTA :10


 UNA CIFRA DEL 1 AL 6 :3


    1    6    6

    A    B    C


 HA PERDIDO

 TOTAL ACTUAL :-20; CONTINUA (S
O N)?
```

El juego de Wytthof

REFERENCIAS: Modalidad: SLOW o FAST.

Dificultad de programación: mediana.

Categoría del juego: de reflexión (para dos jugadores).

Dificultad del juego: mediana.

EL JUEGO: Se dispone de una rejilla cuadrada de 10 × 10 casillas. Aparece un punto fuera de la diagonal, y los jugadores deberán desplazarlo en cada jugada, siguiendo solamente tres direcciones: a la derecha, hacia abajo y en diagonal hacia abajo y a la derecha. El primer jugador que lo consiga en (9, 9) será considerado como ganador, o como perdedor, según el acuerdo tomado al principio.

EL PROGRAMA: Las líneas 1Ø, 2Ø y 3Ø permiten definir una posición de partida, fuera de la diagonal.

Los dos bucles intercalados I (5Ø-9Ø) y J (6Ø-8Ø) dibujan el campo de juego.

La línea 11Ø imprime la posición del punto móvil.

La comprobación de final se efectúa en la línea 115.

El cambio de jugador se asegura con la línea 13Ø.

Las direcciones de desplazamiento disponibles 1, 2 y 3 se descifran mediante tres subprogramas, que empiezan en las líneas 1ØØØ, 2ØØØ y 3ØØØ.

PARA JUGAR: Pulsar RUN para poner en marcha el programa.

El campo de juego se visualiza, así como la posición de salida.

Después se obtiene la visualización JR 1 y abajo la pregunta DIR ¿1, 2, 3?, a la cual se responde tecleando uno de estos tres números, seguido de NEWLINE.

A continuación es el turno del jugador 2.

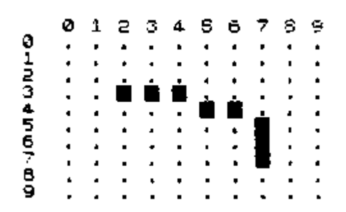
El juego se termina cuando uno de los jugadores llega a la casilla (9, 9).

POSIBLES EXTENSIONES: En primer lugar pueden mejorarse los controles (por ejemplo, la salida del campo de juego no está prohibida), a continuación, mejorar la visualización (grafismo, color), ampliar el juego a varias personas con un campo de juego mayor, introducir un monstruo que intentará capturar a uno de los jugadores, prever un programa que permitirá jugar al ZX, etc.

EL JUEGO DE WYTTHOF

PROGRAMA ZX 81

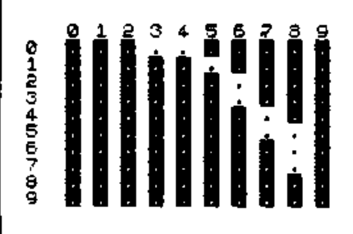
```
  10 LET X=INT (RND*5)
  20 LET Y=INT (RND*5)
  30 IF X=Y THEN GO TO 20
  40 LET A=NOT PI
  50 FOR I=A TO 9
  60 FOR J=A TO 9
  70 PRINT AT INT PI,J+J+INT PI;
J;AT 4+I,A;I;AT 4+I,J+J+INT PI;"
."
  80 NEXT J
  90 NEXT I
 105 LET H=SGN PI
 110 PRINT AT Y+4,X+X+INT PI;"█"
 115 IF X+Y=CODE ")" THEN STOP
 120 PRINT AT CODE ")",SGN PI;"J
UGADOR 2"
 130 IF NOT (H-SGN PI) THEN PRIN
T AT CODE ")",SGN PI;"JUGADOR 1"
 140 PRINT "DIRECCION 1,2,3 ?"
 150 INPUT D
 160 GO SUB D*1000
 170 LET H=-H
 190 GO TO 110
1000 LET X=X+SGN PI
1010 RETURN
2000 GO SUB 1000
3000 LET Y=Y+SGN PI
3010 RETURN
```



```
 JUGADOR 2
DIRECCION 1,2,3 ?
```

PROGRAMA ZX SPECTRUM

```
  10 LET X=INT (RND*5)
  20 LET Y=INT (RND*5)
  30 IF X=Y THEN GO TO 20
  40 PAPER 5: LET A=0
  50 FOR I=0 TO 9
  60 FOR J=0 TO 9
  70 PRINT AT 3,2*J+3;J;AT 4+I,0
;I; INVERSE 1;AT 4+I,2*J+3;"."
  80 NEXT J
  90 NEXT I
 105 LET H=1
 110 PRINT FLASH 1; INK 1;AT Y+4
,X*2+3;".": BEEP 1,0
 115 IF X+Y=18 THEN STOP
 120 PRINT AT 17,1;"JUGADOR 2"
 130 IF NOT (H-1) THEN PRINT AT
17,1;"JUGADOR 1"
 140 PRINT "DIRECCION 1,2,3 ?"
 150 INPUT D
 160 GO SUB D*1000
 170 LET H=-H
 190 GO TO 110
1000 LET X=X+1
1010 RETURN
2000 GO SUB 1000
3000 LET Y=Y+1
3010 RETURN
```



```
 JUGADOR 2
DIRECCION 1,2,3 ?
```

421

REFERENCIAS:

Modalidad: FAST.

Dificultad de programación: fácil.

Categoría del juego: tirar los dados.

Dificultad del juego: mediana.

EL JUEGO:

Cada jugador deberá, por turnos, tirar tres dados y podrá a continuación escoger, en las dos siguientes tiradas, si desea volver a tirar o no 1, 2 o 3 dados. Entonces se marca una puntuación que corresponde al número que indican los dados, siguiendo las reglas establecidas; el siguiente jugador procederá de la misma manera.

EL PROGRAMA:

En primer lugar, se introduce el número de jugadores en la línea 2Ø.

La línea 9Ø visualiza el número del jugador y el número de intentos ya efectuados.

Un subprograma situado en la línea 4ØØØ, que se utiliza 3 veces en el bucle J (14Ø-16Ø), realiza la tirada de los dados.

Los nuevos lanzamientos se efectúan por las líneas 1ØØØ, 2ØØØ, 3ØØØ.

La variable R$ rige la 2.ª y la 3.ª tirada.

La impresión del resultado siempre se realiza por las líneas 1ØØ y 11Ø.

Las comprobaciones del número de intentos y del cambio de jugador se efectúan en las líneas 17Ø y 19Ø.

PARA JUGAR:

Se pone en marcha la ejecución del programa con RUN, lo que lleva a la visualización de NÚM. DE JUGADORES.

Se introduce el número de jugadores y se pulsa NEWLINE. La pantalla se borra y se visualiza JR N.° 1.

INTENTO A, así como el lanzamiento efectuado y los nombres de los dados (A, B, C).

Basta con pulsar NEWLINE si no se desea volver a tirar. Entonces el turno pasa al siguiente jugador. Para realizar otra tirada, bastará con teclear el nombre del o de los dados que quieren volver a tirarse y pulsar NEWLINE; por ejemplo, para volver a tirar A y C, se tecleará AC seguido de NEWLINE.

Los siguientes jugadores procederán de la misma forma.

POSIBLES EXTENSIONES: La más evidente consiste en asegurar la gestión de los resultados de los jugadores, refiriéndose a las reglas usuales de ganancias del juego 421.

Después de cada jugada puede considerarse la visualización de un cuadro recapitulativo de los resultados de los jugadores.

También puede mejorarse la visualización de los dados (remitir a los diversos programas de visualización de dados que se encuentran en esta misma obra).

421

PROGRAMA ZX 81

```
  10 PRINT "NUMERO DE JUGADORES"
  20 INPUT N
  30 LET S=SGN PI
  40 LET I=S
  50 LET K=NOT PI
  60 CLS
  65 LET R$="ABC"
  70 GO TO 140
  80 LET K=K+S
  85 PRINT AT 5,12;"421"
  87 PRINT AT 2,12;"███"
  90 PRINT AT 4,S;"JR. NO. ";I;"
 ;INTENTO ";K
 100 PRINT AT 8,4;A;TAB 8;B;TAB
12;C
 110 PRINT AT 10,4;"A";TAB 8;"B"
;TAB 12;"C"
 120 PRINT AT 15,S;"ABC ?"
 130 INPUT R$
 133 IF R$="" THEN GO TO 180
 137 LET R$=R$+"DDD"
 140 FOR J=S TO INT PI
 145 GO SUB 4000
 150 GO SUB (CODE R$(J)-37)*1000
 160 NEXT J
 170 IF K<INT PI THEN GO TO 80
 180 LET I=I+S
 190 IF I<=N THEN GO TO 50
 200 GO TO 40
1000 LET A=D
1010 RETURN
2000 LET B=D
2010 RETURN
3000 LET C=D
3010 RETURN
4000 LET D=INT (RND*S+S)
4010 RETURN
```

```
          421

JR. NO. 4 ;INTENTO 3


    4   1   1

    A   B   C


ABC ?
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "NUMERO DE JUGADORES"
  20 INPUT N
  30 LET S=1
  40 LET I=S
  50 LET K=0
  60 CLS
  65 LET R$="ABC"
  70 GO TO 140
  80 LET K=K+S
  85 PRINT FLASH 1;AT 5,12;"421"
  87 PRINT INVERSE 1;AT 2,12;"██
█"
  90 PRINT AT 4,S;"JR. NO. ";I;"
 ;INTENTO ";K
 100 PRINT AT 8,4;A: BEEP 1,5*A-
20: PRINT AT 8,8;B: BEEP 1,5*B-2
0: PRINT AT 8,12;C: BEEP 1,5*C-2
0
 110 PRINT INK 4;AT 10,4;"A";AT
10,8;"B";AT 10,12;"C"
 120 PRINT AT 15,S;"ABC ?"
 130 INPUT R$
 133 IF R$="" THEN GO TO 180
 137 LET R$=R$+"DDD"
 140 FOR J=S TO 3
 145 GO SUB 4000
 150 GO SUB (CODE R$(J)-64)*1000
 160 NEXT J
 170 IF K<3 THEN GO TO 80
 180 LET I=I+S
 190 IF I<=N THEN GO TO 50
 200 GO TO 40
1000 LET A=D
1010 RETURN
2000 LET B=D
2010 RETURN
3000 LET C=D
3010 RETURN
4000 LET D=INT (RND*5+S)
4010 RETURN
```

```
          421

JR. NO. 3 ;INTENTO 2


    4   1   2

    A   B   C


ABC ?
```

Rebotes

REFERENCIAS:

Modalidad: SLOW.

Dificultad de programación: fácil.

Categoría del juego: juego de destreza.

Dificultad del juego: mediana.

EL JUEGO:

La idea del juego se basa en la imitación de un billar de bandas.

De un lado a otro de la pantalla se encuentran la bola y el objetivo al que se apunta; una serie de cifras y de letras en lo alto de la pantalla permite escoger el lugar donde se quiere rebotar.

La bola irá de esta manera hacia la parte alta de la pantalla (hacia la parte baja si se introduce un signo menos), rebotará, volverá, si es necesario, a golpear la parte baja de la pantalla, rebotará, y así sucesivamente hasta alcanzar la línea en donde se encuentra el objetivo; si se está suficientemente cerca del objetivo, se ha ganado y se visualizará el número de rebotes.

EL PROGRAMA:

El bucle I (1Ø-3Ø) imprime las cifras y las letras en lo alto de la pantalla.

Las líneas 4Ø y 5Ø definen las posiciones iniciales de la bola y del objetivo.

La línea 6Ø efectúa su impresión.

El símbolo correspondiente a la dirección escogida se introduce en la línea 9Ø.

El bucle I (1ØØ-13Ø) efectúa los desplazamientos de la bola.

La línea 12Ø calcula la nueva posición.

El cambio de dirección, cuando se encuentra una banda, se realiza en la línea 115, por la devolución en una parte independiente del programa, que está situada en las líneas 15Ø a 16Ø.

La línea 14Ø efectúa la impresión del final de la partida.

La línea 155 cuenta el número de rebotes.

PARA JUGAR:

Se pone en marcha el programa con RUN.

Se visualiza el campo de juego, así como las posiciones iniciales de la bola y del objetivo.

Se escoge el lugar donde se quiere hacer rebotar la bola (de 1 a U) y se introduce, seguido de NEWLINE. En este momento, se visualiza la trayectoria, seguida, si existen motivos, de la mención BRAVO; X REBOTES.

Puede realizarse otra partida pulsando una tecla cualquiera, y terminarla en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Será muy interesante, en un aparato que disponga de un grafismo mejor, dibujar un verdadero billar, con su tapiz verde, e indicar la dirección mediante un segmento que simbolice el taco de billar moviéndose alrededor de la bola.

También pueden añadirse agujeros, poner varias bolas de billar, poner sonido a los rebotes o a los contactos con las otras bolas, contabilizar las carambolas con el fin de realizar una partida entre varios jugadores, etc.

REBOTES

PROGRAMA ZX 81

```
   5 CLS
  10 FOR I=NOT PI TO CODE "3"
  20 PRINT CHR$ (I+CODE "0");
  30 NEXT I
  40 LET X=INT (CODE "?"*RND+INT
 PI)
  50 LET Y=INT (CODE "?"*RND+INT
 PI)
  60 PRINT AT X,NOT PI;"O";AT Y,
CODE "3";"█"
  90 INPUT R$
  95 LET S=NOT PI
  97 LET T=X
 100 FOR I=SGN PI TO CODE "3"
 110 LET T=T-X/(CODE R$-CODE "0"
)
 115 IF T>=CODE "=" OR T<=NOT PI
 THEN GO TO 150
 120 PRINT AT T,I;"O"
 130 NEXT I
 140 IF ABS (T-Y)<INT PI THEN PR
INT "BRAVO :";S;" REBOTES"
 145 PAUSE 4E4
 147 RUN
 150 LET X=-X
 155 LET S=S+SGN PI
 160 GO TO 130
```

```
0123456789ABCDEFGHIJKLMNOPQRSTUV

   O O                O O.          O
BRAVO :4 REBOTES                    █

   O    O             O    O        O

 O       O         O        O       O

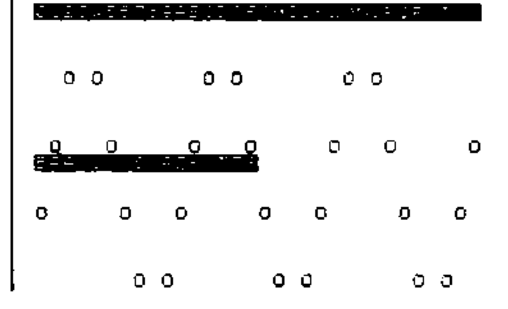
O          O     O           O      O

           O    O           O    O

            O O               O O
```

PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 FOR I=0 TO 31
  20 PRINT INVERSE 1; INK 1;CHR$
 (I+CODE "0") AND I<=9;CHR$ (I+C
ODE "7") AND I>9;
  30 NEXT I
  40 LET X=INT (15*RND+3)
  50 LET Y=INT (15*RND+3)
  60 PRINT AT X,0;"O";AT Y,31;"█
"
  90 INPUT R$
  95 LET S=0
  97 LET T=X
 100 FOR I=1 TO 31
 110 LET T=T-X/(CODE R$-CODE "0"
-7*(R$>="A"))
 115 IF T>=20 OR T<=0 THEN GO TO
 150
 120 PRINT AT T,I;"O"
 130 NEXT I
 140 IF ABS (T-Y)<3 THEN PRINT I
NVERSE 1; INK 2;"BRAVO :";S;" RE
BOTES": BEEP .4,18
 145 PAUSE 4E4
 147 RUN
 150 LET X=-X
 155 LET S=S+1
 160 GO TO 130
```



Jackpot

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: de azar (para un solo jugador).
Dificultad del juego: fácil.

EL JUEGO: Es una variante simplificada de un juego de azar clásico, en el que el jugador debe apostar y tirar de una palanca para poner en marcha el juego. Se gana si se obtienen determinadas combinaciones de símbolos (más o menos, según la apuesta y la combinación de símbolos que se obtenga).

Aquí los símbolos se reemplazan por cifras, y se gana por el hecho de obtener tres cifras idénticas. Existe, además, una combinación especial que permite obtener el JACKPOT.

EL PROGRAMA: El bucle I (4Ø-6Ø) realiza el dibujo de la máquina; la línea 74 visualiza la ubicación de las cifras (vacías).

La apuesta se introduce en la línea 87 y se comprueba en la línea 95: se termina la partida cuando se apuesta Ø (salida hacia la línea 2ØØØ).

Las líneas 13Ø, 15Ø y 17Ø permiten obtener las tres cifras mediante una llamada a un subprograma que se encuentra en la línea 1ØØØ. El resultado se visualiza con la línea 18Ø; después se comprueba en la línea 19Ø y, si se ha ganado, en la línea 31Ø.

PARA JUGAR: Poner en marcha el programa con RUN; cuando se visualiza la pregunta CANTIDAD INICIAL, se responde introduciendo un número suficientemente elevado, a partir del cual se descontarán las apuestas.

Después se pulsa NEWLINE y se introduce una primera apuesta (distinta de Ø), seguida de NEWLINE.

Las tres cifras calculadas se visualizan un breve instante, así como la apuesta realizada y el resultado obtenido (ganancia o pérdida).

A continuación, puede apostarse de nuevo, pulsar NEWLINE, y así sucesivamente.

Para terminar, es suficiente con responder Ø a la pregunta SU APUESTA. En este momento se visualizará el resto de la cantidad inicial.

POSIBLES EXTENSIONES: Podrá considerarse el mejorar sensiblemente los controles, así como utilizar el color y posibilidades gráficas más elaboradas con el fin de simular aún más un auténtico JACKPOT.

JACKPOT

PROGRAMA ZX 81

```
  10 PRINT "CANTIDAD INICIAL"
  20 INPUT P
  30 CLS
  40 FOR I=SGN PI TO INT PI
  50 PRINT AT I+INT PI,VAL "6";"██████████"
  60 NEXT I
  74 PRINT AT VAL "5",VAL "7";"█";TAB VAL "10";"█";TAB VAL "13";"█"
  85 PRINT AT VAL "11",SGN PI;"SU APUESTA"
  87 INPUT M
  95 IF NOT M THEN GO TO 2000
 130 GO SUB 1000
 140 LET A=X
 150 GO SUB 1000
 160 LET B=X
 170 GO SUB 1000
 180 PRINT AT VAL "5",VAL "7";A;TAB VAL "10";B;TAB VAL "13";X
 190 IF A=B AND A=X THEN GO TO 300
 195 LET N=-M
 200 PRINT AT VAL "9",SGN PI;"APUESTA ";M;"  RESULTADO ";N
 204 PAUSE 160
 205 LET P=P+N
 210 GO TO 40
 300 LET N=M*50
 310 IF A=7 THEN LET N=M*500
 320 GO TO CODE "COS "
1000 LET X=INT (VAL "9"*RND+SGN PI)
1010 RETURN
2000 CLS
2010 PRINT "CANTIDAD FINAL ";P;,,"HASTA LUEGO"
```



```
APUESTA 10  RESULTADO -10

SU APUESTA
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "CANTIDAD INICIAL"
  20 INPUT P
  30 CLS
  40 FOR I=1 TO 3
  50 PRINT INK 2;AT I+3,6;"██████████"
  60 NEXT I
  74 PRINT INK 5;AT 5,7;"█";AT 5,10;"█";AT 5,13;"█"
  85 PRINT AT 11,1;"SU APUESTA"
  87 INPUT M
  95 IF NOT M THEN GO TO 2000
 130 GO SUB 1000
 140 LET A=X
 150 GO SUB 1000
 160 LET B=X
 170 GO SUB 1000
 180 PRINT AT 5,7;A: BEEP 1,5*A-20: PRINT AT 5,10;B: BEEP 1,5*B-20: PRINT AT 5,13;X: BEEP 1,5*X-20
 190 IF A=B AND A=X THEN GO TO 300
 195 LET N=-M
 200 PRINT AT 9,1;"APUESTA ";M;"  RESULTADO ";N
 204 PAUSE 160
 205 LET P=P+N
 210 GO TO 40
 300 LET N=M*50
 310 IF A=7 THEN LET N=M*500
 320 GO TO 200
1000 LET X=INT (9*RND+1)
1010 RETURN
2000 CLS
2010 PRINT "CANTIDAD FINAL ";P;,,"HASTA LUEGO"
```



```
APUESTA 10  RESULTADO -10

SU APUESTA
```

El juego del haba

REFERENCIAS: Modalidad: SLOW o FAST.

Dificultad de programación: fácil.

Categoría del juego: de reflexión (para dos jugadores).

Dificultad del juego: mediana.

EL JUEGO: Aunque presenta un cierto parecido con el juego WYTTHOF, esta versión ofrece de hecho cuatro posibilidades de desplazamiento; la posición de comienzo está fijada aquí en (Ø, Ø) y podría, llegado el caso, modificarse.

Las posiciones de desplazamiento disponibles son: a la derecha, hacia abajo, en diagonal hacia abajo y a la derecha, y en diagonal hacia abajo y a la izquierda.

En principio se juega con la regla de que pierde el primero que llega a (9, 9).

EL PROGRAMA: Los bucles intercalados I (2Ø-8Ø) y J (3Ø-7Ø) dibujan el terreno de juego.

La línea 93 efectúa la comprobación del final de la partida.

La línea 125 permite ejecutar uno de los subprogramas de desplazamiento.

Las líneas 13Ø y 132 aseguran el cambio de jugador.

PARA JUGAR: Se pone en marcha el programa con RUN, y se visualiza el terreno de juego, así como la pregunta DIR ¿1, 2, 3, 4?; las cuatro direcciones propuestas corresponden a los desplazamientos siguientes:

El jugador 1 introduce su desplazamiento y pulsa NEWLINE.

Se realiza el movimiento correspondiente, y después se obtiene la visualización JR 2 y DIR ...

El jugador 2 procede de la misma manera, y así sucesivamente, hasta que uno de los jugadores llegue a (9, 9), caso en el cual se termina el juego.

POSIBLES EXTENSIONES: Al principio de la partida pueden introducirse casillas con trampas (cada uno intentará hacer venir al otro a una de estas casillas).

También puede jugarse con varios símbolos, si es posible, de diferentes colores y desplazándolos simultáneamente, considerar terrenos de juego triangulares con dimensiones arbitrarias, etc.

EL JUEGO DEL HABA

PROGRAMA ZX 81

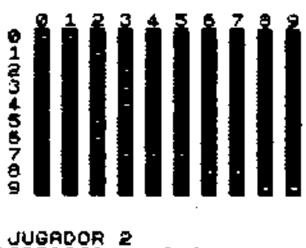
```
  10 LET A=NOT PI
  20 FOR J=A TO 9
  30 FOR I=A TO 9
  40 PRINT AT INT PI,I+I+INT PI;
I,AT J+4,SGN PI;J;AT J+4,I+I+INT
 PI;"*"
  70 NEXT I
  80 NEXT J
  82 LET X=4
  84 LET Y=INT PI
  90 LET J=SGN PI
  96 PRINT AT X,Y;CHR$ (128)
  98 IF X=CODE "$" AND Y=CODE "+
" THEN STOP
 100 PRINT AT 16,SGN PI;"JUGADOR
 ";J
 110 PRINT "DIRECCION 1,2,3,4 ?"
 120 INPUT I
 125 GO SUB I*1000
 130 LET J=J+SGN PI
 132 IF J=INT PI THEN GO TO 90
 140 GO TO 96
1000 LET Y=Y+2
1010 RETURN
2000 GO SUB 1000
3000 LET X=X+SGN PI
3010 RETURN
4000 GO SUB 3000
4010 LET Y=Y-2
4020 RETURN
```

```
   0 1 2 3 4 5 6 7 8 9
 0 █ █ █ * * * * * * *
 1 * * * █ * * * * * *
 2 * * * * █ * * * * *
 3 * * * * █ * * * * *
 4 * * * █ * * * * * *
 5 * * █ * * * * * * *
 6 * * █ * * * * * * *
 7 * * * █ █ █ █ * * *
 8 * * * * * * * █ █ *
 9 * * * * * * * * * █

 JUGADOR 1
DIRECCION 1,2,3,4 ?
```

PROGRAMA ZX SPECTRUM

```
  10 LET A=0
  20 FOR J=0 TO 9
  30 FOR I=0 TO 9
  40 PRINT AT 3,2*I+3;I;AT J+4,1
J; INK 3;AT J+4,2*I+3;"█"
  70 NEXT I
  80 NEXT J
  82 LET X=4
  84 LET Y=3
  90 LET J=1
  96 PRINT INVERSE 1;AT X,Y;"*"
  98 IF X=13 AND Y=21 THEN STOP
 100 PRINT AT 16,1;"JUGADOR ";J
 110 PRINT "DIRECCION 1,2,3,4 ?"
 120 INPUT I
 125 GO SUB I*1000
 130 LET J=J+1
 132 IF J=3 THEN GO TO 90
 140 GO TO 96
1000 LET Y=Y+2
1010 RETURN
2000 GO SUB 1000
3000 LET X=X+1
3010 RETURN
4000 GO SUB 3000
4010 LET Y=Y-2
4020 RETURN
```



Juego de la oca aleatorio

REFERENCIAS:

Modalidad: SLOW (FAST añadiendo una pausa).

Dificultad de programación: difícil.

Categoría del juego: de azar (para dos jugadores).

Dificultad del juego: fácil.

EL JUEGO:

El juego está basado en la vieja idea del juego de la oca, pero añadiendo la posibilidad que ofrece el ordenador de escoger al azar las casillas "con problema".

La regla del juego no tiene gran interés, ya que los jugadores no intervienen en ningún momento durante el desarrollo de la partida, la cual está completamente dirigida por el ZX; basta con pulsar una tecla después de cada tirada de dados y mirar lo que pasa.

EL PROGRAMA:

La línea 5Ø permite llamar al subprograma (1ØØØ a 1Ø6Ø), que realiza la visualización del campo de juego, el control de la posición de los jugadores y la visualización de esta posición. Por otra parte, esta visualización volverá a realizarse durante la partida para tener en cuenta el desplazamiento efectuado realmente (que no es necesariamente el que indica el dado).

La línea 55 se ocupa del cambio de jugador, y la línea 6Ø efectúa la tirada del dado.

La línea 75 comprueba si alguno de los jugadores termina la partida.

La línea 1ØØ calcula un número aleatorio entre Ø y 3, que servirá, en caso necesario, para determinar el verdadero desplazamiento del jugador, enviándolo a uno de los subprogramas de "suerte".

Las líneas 1ØØØ y 1ØØ5 hacen avanzar los peones, y las líneas 1Ø2Ø a 1Ø4Ø vuelven a dibujar, en cada visualización, el campo de juego entero.

PARA JUGAR:

Se pondrá en marcha el juego con RUN. Entonces se dibuja el campo de juego, y después aparece la visualización JR 1 : ..

Deberá pulsarse una tecla cualquiera para tener en cuenta la tirada del jugador número 1; no hay que inquietarse por el tiempo de reacción, que parecerá un poco largo.

El desplazamiento indicado por los datos se ejecuta al cabo de un cierto tiempo; llegado el caso, se efectuará un segundo desplazamiento: se trata del desplazamiento "suerte". Entonces se obtendrá la visualización JR 2 : ..

El jugador 2 pulsará a su vez una tecla cualquiera, y así sucesivamente.

Al final de la partida se dará a conocer el ganador mediante una visualización BRAVO.

Puede pararse en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Puede ampliarse el campo de juego, introducir más riesgos, cambiar su naturaleza.

También podrían escogerse, al principio del juego, las casillas "suerte", añadir visualizaciones en consecuencia, etc.

JUEGO DE LA OCA ALEATORIO

PROGRAMA ZX 81

```
  20 LET X=NOT PI
  30 LET Y=X
  40 LET J=SGN PI
  41 LET U=J+INT PI
  42 LET V=U+U+J+J
  45 LET I=X
  50 GOSUB V*V*V
  55 LET J=ABS (J-SGN PI)
  60 LET I=INT (RND*6+SGN PI)
  75 IF X=CODE "3" OR Y=CODE "3"
 THEN PRINT "BRAVO  ";W
  80 PRINT "JR ";J+SGN PI;"  :";C
HR$ (156+I)
  85 PAUSE V*V*V
  94 GOSUB V*V*V
  97 LET I=INT (U*RND)
  98 GOSUB I*(RND<=.5)*V*V+V*V
 120 GOTO CODE "M"
 200 LET I=-I
 300 LET I=I+I
 310 RETURN
 400 IF NOT J THEN LET X=INT (V*
I*RND)
 410 IF J THEN LET Y=INT (V*I*RN
D)
 420 RETURN
1000 IF NOT J THEN LET X=X+I*((X
+I)<=CODE "3")
1002 LET Y=Y+J*I*(Y+I<=CODE "3")
1008 FOR I=NOT PI TO CODE "3"
1010 PRINT AT INT PI,I;CHR$ (COD
E "0"+I)
1013 PRINT TAB I;CHR$ (CODE "█"+
CODE "1"*(I=X))
1015 PRINT TAB I;CHR$ (CODE "█"+
CODE "2"*(I=Y))
1020 NEXT I
1030 RETURN
```

```
0123456789ABCDEFGHIJKLMNOPQRSTUV

JR 1 :█
```

PROGRAMA ZX SPECTRUM

```
  10 LET X=0
  30 LET Y=X
  40 LET J=1
  45 LET I=X
  50 GO SUB 1000
  55 LET J=ABS (J-1)
  60 LET I=INT (6*RND+1)
  75 IF X=31 OR Y=31 THEN PRINT
"BRAVO ";W
  80 PRINT "JUGADOR ";J+1;"   :";
I: BEEP 0.2,10+10*J
  85 PAUSE 4E4
  90 GO SUB 1000
 100 LET I=INT (4*RND)
 110 GO SUB I*(RND<.5)*50+150
 120 GO TO 50
 150 RETURN
 200 LET I=-I
 210 RETURN
 250 LET I=I+I
 260 RETURN
 300 IF NOT J THEN LET X=INT (30
*RND)
 310 IF J THEN LET Y=INT (30*RND
)
 320 RETURN
1000 IF NOT J THEN LET X=X+I*(X+
I<=31)
1005 LET Y=Y+J*I*(Y+I<=31)
1010 FOR I=0 TO 31
1020 PRINT AT 3,I;I AND I<=9. PR
INT AT 3,I;CHR$ (55+I) AND I>9
1030 PRINT INK 1;AT 4,I;CHR$ (CO
DE "█"-94*(I=X))
1040 PRINT INK 2;AT 5,I;CHR$ (CO
DE "█"-93*(I=Y))
1050 NEXT I
1060 RETURN
```

```
0123456789ABCDEFGHIJKLMNOPQRSTUV

JUGADOR 1   : 6
```

NIVEL 2

N.º	*Nombre del programa*	*Tema y particularidades*
1	**Un dado**	Tirada de un dado, con grafismo.
2	**Visiones**	Un juego sobre las palabras, ¡para el ordenador!
3	**Más grande, más pequeño**	Un juego de deducción interesante, para un jugador.
4	**El electricista loco**	Lógico y difícil, para un solo jugador.
5	**La carrera al 20**	Juego de reflexión, para dos jugadores o más.
6	**Dibujo**	Un método original de codificación.
7	**Dado trucado**	Para utilizar en otros programas, si se es poco honrado.
8	**El juego de la pídola**	Un juego de reflexión para dos jugadores.
9	**Letras**	Un juego de palabras bastante sencillo (dos jugadores).
10	**Espía**	Para ofrecer un ZX a su pareja y declararse en código.
11	**Enganchar los vagones**	Para los más jóvenes, muy visual.
12	**Tirada de un dado trucado**	Para revisar los conocimientos en estadística.
13	**Orden**	Para poner en orden con un poco de astucia lo más rápidamente posible.
14	**Carrera de caballos**	Bastante visual; un juego de azar para varios jugadores.
15	**El juego de Marienbad**	Un juego lógico, para dos jugadores, en una versión ampliada.
16	**Ídem**	Un juego de cálculo mental, para dos jugadores (varios niveles).
17	**Mastermind**	Con cifras, pero muy clásico; un solo jugador.
18	**A través**	Un juego de estrategia elemental para dos jugadores.
19	**Reinas sobre el tablero**	Un viejo problema, para ayudar a reflexionar.
20	**El juego de la vida**	Un gran clásico; dos versiones: manual o automática.

Confección de una tabla Búsqueda de un elemento en una tabla y en una cadena

LA INSTRUCCIÓN DIM, O INSTRUCCIÓN DE CONFECCIÓN DE UNA TABLA

Las estructuras de tablas se utilizan muy corrientemente en programación, demostrándose incluso indispensables en un gran número de casos. Por ejemplo, si se quieren introducir los ingresos y los gastos de todos los días del mes (se supondrán, para simplificar, meses de 30 días), podrían definirse 30 variables llamadas PRIMER DÍA, SEGUNDO DÍA, etc.

Es más fácil y racional utilizar un bucle I, I variante de 1 a 30, y una variable N(I) para los ingresos y G(I) para los gastos; haciendo esto, se ha utilizado, sin saberlo, una estructura de tabla de una dimensión, que no puede ser admitida por el ZX a menos que, previamente, se le haya indicado que iba a utilizarse una tabla de este tipo, mediante una instrucción DIM N(3Ø) para los ingresos y DIM G(3Ø) para los gastos.

Los gastos y los ingresos podrían también mantenerse en una misma tabla, dándole esta vez dos dimensiones: por ejemplo, DIM NG (3Ø,2) [o DIM NG (2,30)]. Así, con DIM NG (3Ø,2), podrá convenirse que NG (I,1) representa los ingresos y NG (I,2) los gastos. También son posibles las tablas de más de dos dimensiones.

Pueden también concebirse tablas en las que los elementos sean cadenas de caracteres y no números, pero entonces es indispensable precisar la longitud máxima de las cadenas autorizadas (es decir, el número máximo de caracteres de cada palabra).

De esta manera, DIM NOM$ (5,1Ø) reserva 5 lugares de la memoria para palabras de 10 letras como máximo.

BÚSQUEDA DE UN ELEMENTO EN UNA TABLA : N(I)

Si se tiene una tabla de una dimensión, definida con una instrucción como DIM N(3Ø), el primer elemento (numérico) de esta tabla se identificará por N(I), y será manejado exactamente como una variable numérica ordinaria (asignación, cálculos, comprobaciones, etcétera).

De esta manera, INPUT N(3) permite introducir a mano el valor del tercer elemento numérico de la tabla N. PRINT N(5) permitirá visualizar el quinto elemento de la misma tabla N, etc.

BÚSQUEDA DE UN ELEMENTO EN UNA CADENA : N$(I)

En el Basic Sinclair, una cadena de caracteres se trata de forma algo parecida a una tabla, que no será útil declarar y que tiene por dimensión el número de caracteres de la cadena.

Esta manera de actuar permite la fácil extracción del primer elemento de la cadena N$, que se denominará simplemente N$ (I); presenta también la ventaja de permitir la fácil sustitución de un elemento de la cadena N$ por una simple asignación.

Por ejemplo: la asignación LET N$ = "BUENOS DÍAS".

PRINT N$(1) visualizará B, PRINT N$(3) visualizará E. Si se escribe Let N$(3) = "U", se obtendrá PRINT N$, que visualizará BUUNOS DÍAS.

Los Basic usuales no permiten esta simplicidad en el tratamiento de las cadenas; la instrucción correspondiente a N$(3) será MID$(N$,3,1); se encontrará, en cambio, más flexibilidad en las tablas de cadenas (difíciles de manejar en el ZX).

Un dado

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: tirada de un dado (con grafismo).
Dificultad del juego: ninguna.

EL JUEGO: El programa simplemente propone la tirada de un dado aleatorio y su representación gráfica en tamaño grande.

Puede fácilmente utilizarse en subprogramas, para todo tipo de juegos que requieran la tirada de uno o varios dados.

EL PROGRAMA: La línea 1Ø utiliza el generador aleatorio del ZX para sacar un entero del 1 al 6.

El bucle J (4Ø-6Ø) dibuja los bordes del dado, y las líneas 30 y 65 las partes inferior y superior.

Las tres líneas 7Ø, 8Ø y 9Ø se utilizan para el dibujo de los puntos.

La línea 12Ø permite, en caso necesario, volver a tirar.

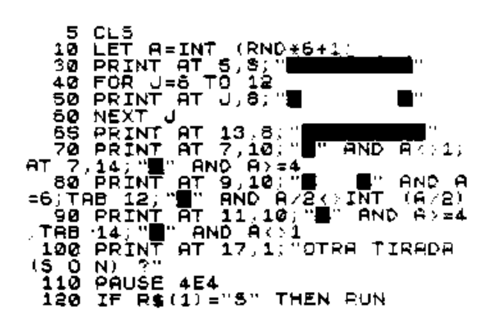
PARA JUGAR: Se tira el dado con RUN; el resultado se visualiza en forma simbólica, como en un dado real.

A continuación se visualiza la pregunta OTRA TIRADA (S o N). Si se contesta S, se efectuará otro lanzamiento.

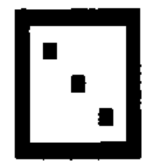
POSIBLES EXTENSIONES: Este programa podrá utilizarse como subprograma cada vez que se necesite simular una tirada de dados.

UN DADO

PROGRAMA ZX 81

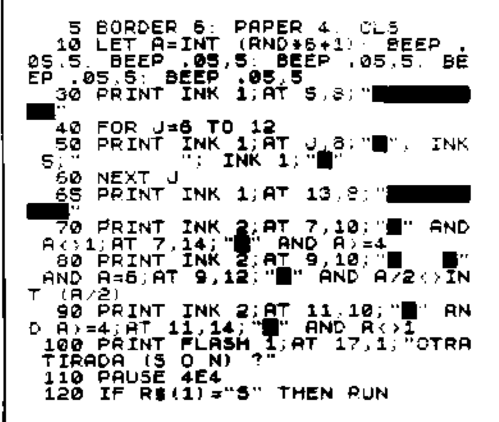


```
   5 CLS
  10 LET A=INT (RND*6+1)
  30 PRINT AT 5,8;"████████"
  40 FOR J=6 TO 12
  50 PRINT AT J,8;"█      █"
  60 NEXT J
  65 PRINT AT 13,8;"████████"
  70 PRINT AT 7,10;"█" AND A<>1;
AT 7,14;"█" AND A>=4
  80 PRINT AT 9,10;"█   █" AND A
=6;TAB 12;"█" AND A/2<>INT (A/2)
  90 PRINT AT 11,10;"█" AND A>=4
;TAB 14;"█" AND A<>1
 100 PRINT AT 17,1;"OTRA TIRADA
(S O N) ?"
 110 PAUSE 4E4
 120 IF R$(1)="S" THEN RUN
```

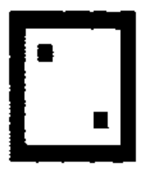


OTRA TIRADA (S O N) ?

PROGRAMA ZX SPECTRUM



```
   5 BORDER 6: PAPER 4: CLS
  10 LET A=INT (RND*6+1): BEEP .
05,5: BEEP .05,5: BEEP .05,5: BE
EP .05,5: BEEP .05,5
  30 PRINT INK 1;AT 5,8;"███████
██"
  40 FOR J=6 TO 12
  50 PRINT INK 1;AT J,8;"█"; INK
 5;"       "; INK 1;"█"
  60 NEXT J
  65 PRINT INK 1;AT 13,8;"███████
██"
  70 PRINT INK 2;AT 7,10;"█" AND
 A<>1;AT 7,14;"█" AND A>=4
  80 PRINT INK 2;AT 9,10;"█   █"
 AND A=6;AT 9,12;"█" AND A/2<>IN
T (A/2)
  90 PRINT INK 2;AT 11,10;"█" AN
D A>=4;AT 11,14;"█" AND A<>1
 100 PRINT FLASH 1;AT 17,1;"OTRA
 TIRADA (S O N) ?"
 110 PAUSE 4E4
 120 IF R$(1)="S" THEN RUN
```



OTRA TIRADA (S O N) ?

Visiones

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: dibujo.
Dificultad del juego: fácil.

EL JUEGO: ¡El ZX permite, por primera vez, entrar en su intimidad, dejando entrever su propia representación del mundo!

Se le propondrán palabras, no importa cuáles, y él contestará a su vez presentando una representación gráfica de lo que significan estas palabras.

El juego será mucho más divertido si lo practican varias personas, y puede que a veces se lleven alguna sorpresa.

EL PROGRAMA: Un bucle I (4Ø-12Ø) se encarga de realizar los dibujos.

La línea 5Ø permite transformar en número cada letra de la palabra que se introduce al principio (línea 2Ø).

Este número será traducido a continuación en base 2 por un bucle J (líneas 7Ø a 11Ø); según el resultado de esta transformación, se imprimirá un punto o no (línea 9Ø).

PARA JUGAR: Se inicia la ejecución del programa con RUN, lo que lleva a visualizar la pregunta QUÉ DIBUJO. Se responde introduciendo una palabra cualquiera, la cual será transformada en dibujo por el ZX después de pulsar NEWLINE.

Para volver a empezar, será suficiente con pulsar una tecla cualquiera, y se terminará con BREAK.

POSIBLES EXTENSIONES: Puede seguirse la misma idea introduciendo elementos de simetría en los dibujos y añadiéndoles color.

Tambien puede imaginarse por este método la codificación de un cierto número de dibujos, que se introducirían en la máquina uno detrás de otro, con el fin de crear un juego en el que el ZX se volverá cada vez más habilidoso. También puede efectuarse un juego para dos personas, en el que el ZX pide encontrar la palabra que corresponda a un determinado dibujo, etc.

VISIONES

PROGRAMA ZX 81

```
 10 PRINT AT SGN PI,NOT PI;"QUE
DIBUJO :";
 20 INPUT R$
 30 PRINT R$
 40 FOR I=SGN PI TO LEN R$
 50 LET N=9*(CODE R$(I))-340
 60 LET J=SGN PI
 70 LET S=(N/2)-INT (N/2)
 80 LET N=INT N/2
 90 IF S THEN PRINT AT I+5,J+12
"█"
100 LET J=J+SGN PI
110 IF J-9 THEN GO TO 70
120 NEXT I
130 PAUSE 4E4
150 CLS
160 GO TO 10
```

```
QUE DIBUJO :CUADRO
```



PROGRAMA ZX SPECTRUM

```
  10 PRINT AT 1,0;"QUE DIBUJO :"
;
  20 INPUT R$
  30 PAPER SQR ((LEN (R$))*RND+1
): CLS : PRINT R$
  40 FOR I=1 TO LEN R$
  50 LET N=9*(CODE R$(I)-27)-340
  60 LET J=1
  70 LET S=(N/2)-INT (N/2)
  80 LET N=INT N/2
  90 IF S THEN PRINT INK 9;AT I+
5,J+12;"█"
 100 LET J=J+1
 110 IF J-9 THEN GO TO 70
 120 NEXT I
 130 PAUSE 4E4
 150 BORDER 7*RND: CLS
 160 GO TO 10
```

```
JACQUES
```



Más grande, más pequeño

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: fácil.

Categoría del juego: juego de deducción (para un solo jugador).

Dificultad del juego: difícil.

EL JUEGO: Se inspira en un principio ligeramente distinto al del MASTERMIND. Se trata igualmente de adivinar un número (de 1 a 7 cifras), pero las indicaciones de que se dispone son esta vez, de una parte, el número de cifras que en el número probado son superiores a las del número que ha de adivinarse y, por otra parte, el número de cifras que son inferiores. Si, por ejemplo, el número que tiene que adivinarse es el 3054 y se prueba el 4251, la máquina responderá 2 más grandes (corresponden al 4 y al 2) y uno más pequeño (el 1).

EL PROGRAMA: La línea 3Ø no permite pedir números de más de 8 cifras.

El número que ha de adivinarse está situado en la línea 5Ø, en una variable alfanumérica X$.

Las comparaciones con el número probado se efectúan en la línea 95 (que comprueba si es exacto) y en el bucle I (12Ø-17Ø), donde se realiza la comparación "palabra por palabra".

La visualización del resultado se realiza gracias a las líneas 18Ø y 19Ø.

El contador del número de intentos está en la línea 195.

PARA JUGAR: La iniciación del programa con RUN provoca la visualización de NÚM. DE CIFRAS.

Se introduce un número inferior a 8 y se pulsa la tecla NEWLINE.

Se visualiza el número de intentos, así como la pregunta NÚM. SUPUESTO; se responde introduciendo un número seguido de NEWLINE.

Después, el resultado se visualiza bajo la forma ...CIFRAS MÁS GRANDES y ...CIFRAS MÁS PEQUEÑAS.

Si el número no es exacto, se vuelve a empezar (teniendo cuidado de introducir solamente números que tengan las mismas cifras que las que deben adivinarse).

Si se ha encontrado el número, el ZX visualizará ENCONTRADO ... EN ... INTENTOS. Después, OTRA PARTIDA ¿(S o N)?; se tecleará S, seguido de NEWLINE, para volver a empezar una nueva partida.

POSIBLES EXTENSIONES: Como en el MASTERMIND, se puede imaginar un juego parecido con letras, colores o también utilizando otros símbolos.

También puede preverse dejar jugar al ZX, el cual tendrá entonces el papel de descifrador.

MÁS GRANDE, MÁS PEQUEÑO

PROGRAMA ZX 81

```
  5 CLS
 10 PRINT "NUMERO DE CIFRAS ";
 20 INPUT C
 30 IF C>8 THEN RUN
 40 PRINT C
 45 LET E=SGN PI
 50 LET X$=STR$ (INT (RND*9**C+
10**(C-E)))
 60 PRINT AT 5,1;"INTENTO NO ";
E
 70 PRINT AT 8,1;"NUMERO SUPUES
TO ";
 80 INPUT P$
 90 PRINT P$
 95 IF P$=X$ THEN GO TO 200
100 LET A=NOT PI
110 LET B=A
120 FOR I=SGN PI TO C
140 IF P$(I)>X$(I) THEN LET A=A
+SGN PI
150 IF P$(I)<X$(I) THEN LET B=B
+SGN PI
170 NEXT I
180 PRINT AT 13,1;A;" CIFRAS MA
S GRANDES"
190 PRINT AT 15,1;B;" CIFRAS MA
S PEQUENAS"
195 LET E=E+SGN PI
197 GO TO 60
200 CLS
210 PRINT AT 5,1;"ENCONTRADO ";
X$;" EN ";E;" INTENTOS"
220 PRINT AT 8,1;"OTRA PARTIDA
(S O N) ?"
230 INPUT R$
240 IF R$="S" THEN RUN
```

```
NUMERO DE CIFRAS 4


 INTENTO NO 4

 NUMERO SUPUESTO  567


 2 CIFRAS MAS GRANDES

 2 CIFRAS MAS PEQUENAS
```

PROGRAMA ZX SPECTRUM

```
  5 CLS
 10 PRINT "NUMERO DE CIFRAS ";
 20 INPUT C
 30 IF C>8 THEN RUN
 40 PRINT C
 45 LET E=1
 50 LET X$=STR$ (INT (RND*9↑C+1
0↑(C-E)))
 60 PRINT INVERSE 1;AT 5,1;"INT
ENTO NO "; FLASH 1;E
 70 PRINT AT 8,1;"NUMERO SUPUES
TO ";
 80 INPUT P$
 90 PRINT P$
 95 IF P$=X$ THEN GO TO 200
100 LET A=0
110 LET B=A
120 FOR I=1 TO C
140 IF P$(I)>X$(I) THEN LET A=A
+1
150 IF P$(I)<X$(I) THEN LET B=B
+1
170 NEXT I
180 PRINT PAPER 4;AT 13,1;A;" C
IFRAS MAS GRANDES"
190 PRINT PAPER 5;AT 15,1;B;" C
IFRAS MAS PEQUENAS"
195 LET E=E+1: BEEP 0.2,10
197 GO TO 60
200 CLS
210 PRINT AT 5,1;"ENCONTRADO ";
X$;" EN ";E;" INTENTOS"
220 PRINT AT 8,1;"OTRA PARTIDA
(S O N) ?"
230 INPUT R$
240 IF R$="S" THEN RUN
```

```
NUMERO DE CIFRAS 4


 INTENTO NO 3

 NUMERO SUPUESTO 3567


 2 CIFRAS MAS GRANDES

 2 CIFRAS MAS PEQUENAS
```

El electricista loco

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: difícil.
Categoría del juego: solitario.
Dificultad del juego: bastante difícil.

EL JUEGO: Una casa con 10 ventanas numeradas del Ø al 9 aparece en la pantalla.

El jugador deberá, en cada intento, manejar uno de los interruptores indicando su número (del Ø al 9).

En principio, la ventana correspondiente debería apagarse, pero también es posible que otra se encienda (o se apague). Se trata, por supuesto, de conseguir que se apaguen las 10 ventanas con el menor número de intentos.

EL PROGRAMA: Una tabla A, en la línea 15, contendrá el estado (encendido o apagado) de las 10 ventanas.

Los bucles I (2Ø-5Ø) y L (62-7Ø) dibujan la casa en su estado inicial.

La línea 85 examina si todo está apagado, en cuyo caso se bifurcará a la línea 229 para visualizar el número de intentos.

La línea 1ØØ espera la indicación del interruptor que se ha operado.

Las líneas 11Ø a 18Ø definen el resultado de la operación.

PARA JUGAR: RUN provoca la visualización de la casa con todas las ventanas encendidas.

A la pregunta INTER se contesta indicando el número de interruptor que se utilizará, seguido de NEWLINE. El resultado de la operación se visualiza entonces en la pantalla (las ventanas se encienden o se apagan).

Se continúa así hasta que todas las ventanas se hayan apagado, obteniéndose entonces la visualización del número de intentos realizados.

POSIBLES EXTENSIONES: Puede mejorarse el grafismo con la utilización, si es posible, del color. También puede definirse un juego para dos personas, aumentar el número de ventanas, modificar la fórmula que define el encendido o el apagado, etc.

EL ELECTRICISTA LOCO

PROGRAMA ZX 81

```
  10 LET S=SGN PI
  12 LET M=NOT PI
  15 DIM A(CODE "▒")
  20 FOR I=S TO CODE "*"
  30 PRINT AT NOT PI,I;"■",AT S,
I;"■";AT S+S,I;"■";AT INT PI,I;"
■";AT INT PI+S,I;"■"
  50 NEXT I
  60 LET T=NOT PI
  62 FOR I=S TO CODE "▒"
  64 LET T=T+A(I)
  65 PRINT AT S+S,I+I+S;"■"
  70 PRINT AT S+S,I+I+S;" " AND
A(I)=NOT PI;AT INT PI+S,I+I+S;CH
R$ (155+I)
  80 NEXT I
  85 IF T=CODE "▒" THEN GOTO COD
E " FAST "
  90 PRINT AT CODE "£",S;"INTER
-";
 100 INPUT N
 110 LET N=N+S
 120 LET M=M+S
 130 IF A(N)=S THEN GOTO CODE "C
OS "
 140 LET A(N)=S
 150 LET R=TAN (RND+N/RND-N)-SIN
 (RND/N)+336*SIN (CODE "▒"*N)
 160 LET N=INT (CODE "▒"*(R-INT
R))
 165 IF NOT N THEN LET N=INT (S+
CODE "▒"*RND)
 170 IF A(N)=S THEN GOTO CODE "C
OS "
 180 LET A(N)=S
 190 GOTO CODE "U"
 200 LET A(N)=NOT PI
 210 GOTO CODE "■"
 229 PRINT AT CODE "£",S;M;" INT
ENTOS"
```



INTER

PROGRAMA ZX SPECTRUM

```
  10 LET S=1
  12 LET M=0
  15 DIM A(10)
  20 FOR I=S TO 23
  30 PRINT INK 5;AT 0,I,"■";AT S
,I;"■";AT S+S,I;"■";AT 3,I;"■";A
T 4,I;"■"
  50 NEXT I
  60 LET T=0
  62 FOR I=S TO 10
  64 LET T=T+A(I)
  65 PRINT INK 5;AT S+S,2*I+S,"■
"
  70 PRINT PAPER 2;AT S+S,2*I+S;
" " AND A(I)=0;AT 4,2*I+S; PAPER
 3; BRIGHT 1;I-1
  80 NEXT I
  85 IF T=10 THEN GO TO 229
  90 PRINT AT 12,S;"NUMERO DE IN
TERRUPTOR :    ";
 100 INPUT N: BEEP 1,10: PRINT F
LASH 1; INK 3;AT 12,25;N
 110 LET N=N+S
 120 LET M=M+S
 130 IF A(N)=S THEN GO TO 200
 140 LET A(N)=S
 150 LET R=TAN (RND+N/RND-N)-SIN
 (RND/N)+336*SIN (8*N)
 160 LET N=INT (10*(R-INT R))
 165 IF NOT N THEN LET N=INT (5+
10*RND)
 170 IF A(N)=S THEN GO TO 200
 180 LET A(N)=S
 190 GO TO 60
 200 LET A(N)=0
 210 GO TO 150
 229 PRINT FLASH 1;AT 15,5;"ENCO
NTRADO EN ";M;" INTENTOS"
```



NUMERO DE INTERRUPTOR : 7

La carrera al 20

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: fácil.
Categoría del juego: lógica (para dos jugadores o más).
Dificultad del juego: mediana.

EL JUEGO: Para este juego pueden idearse varias reglas, pero aquí sólo hemos considerado la regla normal para dos jugadores, que hace que sea un juego parecido a los de desplazamiento o de cerillas: 4 montones de 20 estrellas aparecen al inicio de la partida, y cada jugador puede coger tantas estrellas como quiera, pero en un solo montón a la vez. Se considera perdedor el que coja la última estrella.

EL PROGRAMA: El bucle I (3Ø-5Ø) dibuja 4 filas de 20 estrellas.
La variable F, dimensionada en la línea 6Ø, servirá para registrar el número de estrellas cogidas en cada línea.
La variable T contendrá el número total de estrellas sacadas.
El bucle I (7Ø-9Ø) imprime los números de línea.
La línea 125 controla que existan suficientes estrellas en la línea indicada.
Los bucles I (17Ø-21Ø) y J (18Ø-2ØØ) borran las estrellas; la línea 24Ø comprueba el final de la partida.

PARA JUGAR: Iniciar el programa con RUN; se visualiza el campo de juego, y después aparecen las inscripciones JR 1 y ¿LÍNEA, CASILLAS? Se responde introduciendo un número de línea (entre 1 y 4), seguido de NEWLINE, y del número de estrellas que se deseen sacar, seguido de NEWLINE. En este momento se borran las estrellas y es el turno del segundo jugador.
La partida termina cuando se quita la última estrella, con el anuncio del jugador que ha ganado.

POSIBLES EXTENSIONES: Además de una mejora en la calidad del gráfico y de los controles, pueden considerarse numerosas variantes, modificando, si es necesario, las reglas de salida: puede autorizarse a sacar solamente un número impar de estrellas, de alternar par e impar, decidir al azar el número de estrellas a sacar o jugar varias personas, etc.

LA CARRERA AL 20

PROGRAMA ZX 81

```
  10 LET A=SGN PI
  30 FOR I=A TO 20
  40 PRINT AT INT PI,I+INT PI;"*
";AT 5,I+INT PI;"*";AT 7,I+INT P
I;"*";AT 9,I+INT PI;"*"
  50 NEXT I
  55 LET T=NOT PI
  60 DIM F(4)
  70 FOR I=A TO 4
  75 PRINT AT I+I+A,A;I
  80 LET F(I)=NOT PI
  90 NEXT I
  95 LET B=A
 100 PRINT AT 12,A;"JUGADOR ";B
 110 PRINT AT 14,A;"LINEA,CASILL
AS?"
 120 INPUT L
 123 INPUT C
 125 IF C+F(L)>CODE "=" THEN GO
TO 110
 165 LET F(L)=F(L)+C
 167 LET T=T+C
 170 FOR I=A TO 4
 180 FOR J=NOT PI TO F(I)
 190 PRINT AT I+I+A,J+INT PI;" "
 200 NEXT J
 210 NEXT I
 220 LET B=B+A
 230 IF B=INT PI THEN LET B=A
 240 IF T<80 THEN GO TO 100
 300 CLS
 310 PRINT "EL JUGADOR ";B;" HA
GANADO"
```

```
1    ******************

2       ***************

3  ********************

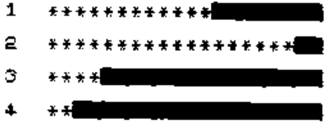
4          ************

JUGADOR 2

LINEA,CASILLAS?
```

PROGRAMA ZX SPECTRUM

```
  10 LET A=1
  30 FOR I=A TO 20
  40 PRINT INK 2;AT 3,I+3;"█"; I
NK 3;AT 5,I+3;"█"; INK 4;AT 7,I+
3;"█"; INK 5;AT 9,I+3;"█"
  50 NEXT I
  55 LET T=0
  60 DIM F(4)
  70 FOR I=A TO 4
  75 PRINT AT 2*I+A,A;I
  80 LET F(I)=0
  90 NEXT I
  95 LET B=A
 100 PRINT AT 12,A;"JUGADOR ";B
 110 PRINT AT 14,A;"LINEA,CASILL
AS?"
 120 INPUT L: IF L>4 THEN GO TO
110
 123 INPUT C
 125 IF C+F(L)>20 THEN GO TO 110
 165 LET F(L)=F(L)+C
 167 LET T=T+C
 170 FOR I=1 TO 4
 180 FOR J=1 TO F(I)
 190 PRINT BRIGHT 1; PAPER I+1;A
T 2*I+A,J+3;"*"
 200 NEXT J
 210 NEXT I
 220 BEEP 1,10: LET B=B+A
 230 IF B=3 THEN LET B=A
 240 IF T<80 THEN GO TO 100
 300 CLS
 310 PRINT "EL JUGADOR ";B;" HA
GANADO": BEEP 1,0
```



```
JUGADOR 2

LINEA,CASILLAS?
```

Dibujo

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: fácil.
Categoría del juego: dibujo.
Dificultad del juego: mediana.

EL JUEGO: Es sobre todo un método algo original de codificación de un dibujo, que tiene la ventaja de poder utilizarse en algunos programas.

EL PROGRAMA: La codificación del dibujo, en el programa retenido, está situada en una variable A$, que está definida en la línea 1Ø.

Llegado el caso, se podrán intentar otros dibujos introduciendo directamente sus códigos: basta con modificar la línea referida por 1Ø INPUT A$.

El bucle N (2Ø-5Ø) descifra la variable A$.

La línea 4Ø determina las posiciones del principio de la escritura.

La línea 45 detecta el carácter Ø, que sirve para indicar el retorno a la línea.

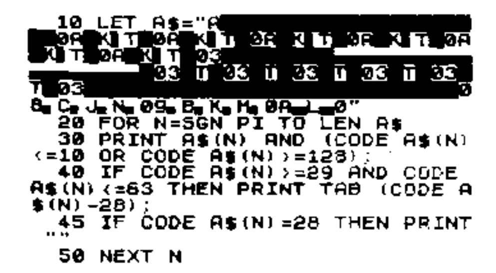
PARA JUGAR: Deberá haberse definido previamente el dibujo, por una variable A$, utilizando los caracteres gráficos y proporcionando la posición de impresión de cada carácter con respecto al principio de la línea, con una cifra de 1 a 9 o con una letra (A : 1Ø ; B : 11 ; C : 12 ; etc.).

El carácter Ø indicará el paso a la línea siguiente.

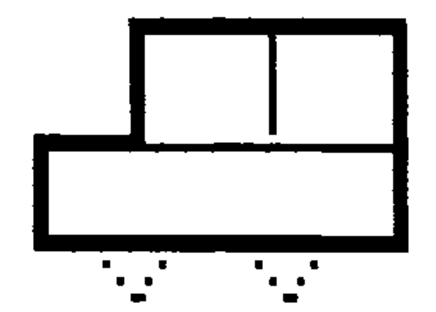
POSIBLES EXTENSIONES: La función PLOT permitirá definir un gráfico más correcto; también podrán considerarse algunas indicaciones para el color, imaginar un juego en el que cada jugador intentaría codificar un dibujo dado lo más rápidamente posible, etc.

DIBUJO

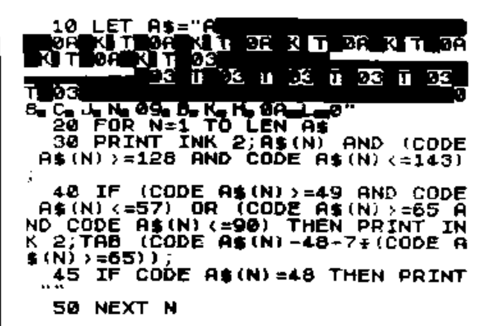
PROGRAMA ZX 81



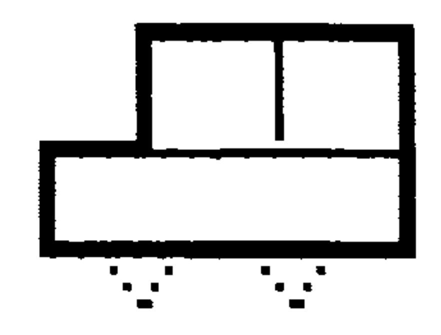
```
  10 LET A$="A
 0A K T 0A K T 0A K T 0A K T 0A
K T 0A K T 03
           03 T 03 T 03 T 03 T 03
T 03                          0
8 C J N 09 B K H 0A L 0"
  20 FOR N=SGN PI TO LEN A$
  30 PRINT A$(N) AND (CODE A$(N)
<=10 OR CODE A$(N)>=128);
  40 IF CODE A$(N)>=29 AND CODE
A$(N)<=63 THEN PRINT TAB (CODE A
$(N)-28);
  45 IF CODE A$(N)=28 THEN PRINT
""
  50 NEXT N
```



PROGRAMA ZX SPECTRUM



```
  10 LET A$="A
 0A K T 0A K T 0A K T 0A K T 0A
K T 0A K T 03
           03 T 03 T 03 T 03 T 03
T 03                          0
8 C J N 09 B K H 0A L 0"
  20 FOR N=1 TO LEN A$
  30 PRINT INK 2;A$(N) AND (CODE
 A$(N)>=128 AND CODE A$(N)<=143)
;
  40 IF (CODE A$(N)>=49 AND CODE
 A$(N)<=57) OR (CODE A$(N)>=65 A
ND CODE A$(N)<=90) THEN PRINT IN
K 2;TAB (CODE A$(N)-48-7*(CODE A
$(N)>=65));
  45 IF CODE A$(N)=48 THEN PRINT
""
  50 NEXT N
```



Dado trucado

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: fácil.
Categoría del juego: tirar los dados.
Dificultad del juego: fácil.

EL JUEGO: Se intentará fabricar un dado trucado según la propia conveniencia. El método que aquí se propone consiste en definir al principio una probabilidad de realización para cada tipo de tirada (es decir, la probabilidad de obtener un 1, un 2, un 3, etc., que cada una de las probabilidades estén comprendidas entre Ø y 1 y que la suma de todas las probabilidades sea un número muy cercano a 1).

EL PROGRAMA: Utiliza el generador aleatorio del SINCLAIR ZX, que se supone de calidad conveniente (si es necesario, podrá utilizarse fácilmente cualquier otro generador).

El bucle I (2Ø-95) se ocupa de la entrada de cada probabilidad, con verificación de cada entrada y del total (línea 6Ø); cada probabilidad se guarda en una variable indexada E (I) : línea 4Ø del programa.

La tirada del dado trucado se realiza con la ayuda de otro bucle I (12Ø-14Ø), y la línea 13Ø permite trucar el dado.

El resultado se visualiza gracias a la línea 15Ø.

Podrá realizarse nueva tirada con la simple pulsación de una tecla (líneas 16Ø y 17Ø).

PARA JUGAR: Después de la puesta en marcha de la ejecución con RUN, se obtiene la visualización de la pregunta PROBABILIDAD DEL DADO 1, a la cual se responde con un número comprendido entre Ø y 1. Es conveniente no olvidar que el total de todas las probabilidades introducidas no deberá en ningún caso pasar de 1.1 (se tolera Ø.1). Podrán visualizarse mensajes de error, y llegado el caso podrá volver a iniciarse el programa.

Cuando todas las probabilidades estén definidas, se efectuará automáticamente la primera tirada, y bastará con pulsar una tecla cualquiera (excepto BREAK) para realizar las tiradas siguientes. Se termina el programa pulsando BREAK.

POSIBLES EXTENSIONES: Evidentemente será muy tentador utilizar astutamente este pequeño programa para mejorar las posibilidades de ganar en un juego de azar en el que se tiren los dados. Entonces podrá considerarse la fijación de los datos que definen el dado trucado en una variable A$, etc.

También podrá considerarse la presentación de los resultados correspondientes a una serie de tiradas de dados bajo una forma espectacular, que muestre el trucado de los dados.

DADO TRUCADO

PROGRAMA ZX 81

```
  10 LET P=NOT PI
  15 DIM E(6)
  20 FOR I=SGN PI TO 6
  30 PRINT AT INT PI,NOT PI;"PRO
BABILIDAD DEL DADO ";I
  40 INPUT E(I)
  50 LET P=P+E(I)
  60 IF E(I)<NOT PI OR E(I)>SGN
PI OR P>1.1 THEN GO TO 200
  70 PRINT AT NOT PI,5*I-5;"DADO
 ";I;AT SGN PI,5*I-5;E(I)
  80 NEXT I
  90 IF P<.9 THEN RUN
  95 PAUSE 60
 110 LET N=INT (RND*6+SGN PI)
 120 FOR I=SGN PI TO 6
 130 IF N=I AND RND>E(I) THEN GO
 TO 110
 140 NEXT I
 150 PRINT AT INT PI,NOT PI;"TIR
ADA :";N;"
                               "
 160 PAUSE 4E4
 170 GO TO 110
 200 PRINT "IMPOSIBLE : 0 <= P <
= 1"
 210 LET P=P-E(I)
 215 PAUSE 50
 220 PRINT AT 4,NOT PI,"
                  "
 230 GO TO 30
```

```
DADO DADO DADO DADO DADO DADO 6
0.1  0.3  0.2  .05  .05  0.3

TIRADA :1
```

PROGRAMA ZX SPECTRUM

```
  10 LET P=0
  15 DIM E(6)
  20 FOR I=1 TO 6
  30 PRINT FLASH 1;AT 3,0;"PROBA
BILIDAD DEL DADO ";I
  40 INPUT E(I)
  50 LET P=P+E(I)
  60 IF E(I)<0 OR E(I)>1 OR P>1.
1 THEN GO TO 200
  70 PRINT PAPER 5;AT 0,5*I-5;"D
ADO ";I;AT 1,5*I-5;E(I)
  80 NEXT I
  90 IF P<.9 THEN RUN
  95 PAUSE 60
 100 RANDOMIZE
 110 LET N=INT (RND*6+1)
 120 FOR I=1 TO 6
 130 IF N=I AND RND>E(I) THEN GO
 TO 110
 140 NEXT I
 150 PRINT PAPER 2; BRIGHT 1; IN
K 4;AT 3,0;"TIRADA "; INVERSE 1;
 INK 1;"          "; FLASH 1; INK 7;
"  ";N;"  "; INK 1; FLASH 0;"
   "; BEEP 1,7*N-24
 160 PAUSE 4E4
 170 GO TO 110
 200 PRINT FLASH 1;"IMPOSIBLE :
0 <= P <= 1": BEEP .3,0: BEEP .3
,0: BEEP .3,0
 210 LET P=P-E(I)
 215 PAUSE 50
 220 PRINT AT 4,0;"
                  "
 230 GO TO 30
```

```
DADO DADO DADO DADO DADO DADO 6
0.2  0.3  0.2  0.1  0.1  0.1

TIRADA [redacted]
```

El juego de la pídola

REFERENCIAS: Modalidad: SLOW o FAST.

Dificultad de programación: fácil.

Categoría del juego: de reflexión (para dos jugadores).

Dificultad del juego: fácil.

EL JUEGO: A pesar de que la idea es muy simple, no es necesario descubrir una estrategia conveniente. Se dispone de tres ovejas negras y de tres ovejas grises, representadas por cuadrados, que ocupan 8 casillas; las ovejas pueden desplazarse de una casilla a otra libre. Podrán intervenir diversas reglas, según el propio interés por una u otra; se obliga a las ovejas a colocarse en la primera casilla disponible, saltando, llegado el caso, por encima de las ovejas contrarias, o bien se las deja escoger la casilla de llegada, lo que puede hacer las partidas interminables.

EL PROGRAMA: El bucle I (5Ø-8Ø) inicializa una tabla A(I) que contiene las posiciones de las ovejas.

El bucle I (1ØØ-14Ø) dibuja el terreno de juego y las ovejas.

Las líneas 15Ø a 19Ø se ocupan del texto y de las entradas.

La línea 195 controla la validez de la jugada.

La línea 22Ø realiza el cambio de jugador.

PARA JUGAR: Se empieza la ejecución del programa con RUN. Entonces se obtiene la visualización JR1, que indica el número de jugador, seguida de la pregunta ¿PRINCIPIO, FIN?

Se contesta indicando el número del peón que ha de desplazarse, seguido de NEWLINE, y después el número de la casilla de llegada, seguida de NEWLINE. En este momento se efectúa el desplazamiento; el segundo jugador procederá de la misma forma.

En principio, el vencedor será el que consiga situar a sus tres ovejas, una al lado de otra, en los lugares que inicialmente ocupaban las de su adversario.

POSIBLES EXTENSIONES: Se empezará por añadir un control al jugador que gana.

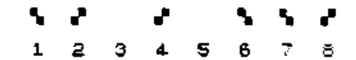
Igualmente puede mejorarse la presentación, si se dibujan ovejas auténticas, numerándolas para que el juego sea más atractivo e imponiendo un determinado orden de llegada.

Por último, podrá definirse una estrategia para el ordenador, fijando las reglas.

EL JUEGO DE LA PÍDOLA

PROGRAMA ZX 81

```
  10 LET A=SGN PI
  20 DIM A(8)
  30 LET A(SGN PI)=NOT A
  40 LET A(8)=NOT A
  50 FOR I=A TO INT PI
  60 LET A(A+I)=A
  70 LET A(4+I)=-A
  80 NEXT I
  90 LET H=A
 100 FOR I=A TO 8
 110 PRINT AT 7,INT PI*I+A;I
 115 PRINT AT 5,INT PI*I+A;" "
 120 IF A(I)=A THEN PRINT AT 5,I
NT PI*I+A;"▞"
 130 IF A(I)=-A THEN PRINT AT 5,
INT PI*I+A;"▚"
 140 NEXT I
 150 PRINT AT 11,SGN PI;"JUGADOR
 2"
 160 IF NOT (H-A) THEN PRINT AT
11,SGN PI;"JUGADOR 1"
 170 PRINT "PRINCIPIO , FIN ?"
 180 INPUT D
 190 INPUT F
 195 IF A(D)<>H OR A(F)<>NOT A T
HEN GO TO 150
 200 LET A(D)=NOT A
 210 LET A(F)=A*H
 220 LET H=-H
 230 GO TO 100
```



```
 JUGADOR 1
PRINCIPIO , FIN ?
```

PROGRAMA ZX SPECTRUM

```
  10 LET A=1: PAPER 7: CLS
  20 DIM A(8)
  30 LET A(1)=0
  40 LET A(8)=0
  50 FOR I=A TO 3
  60 LET A(A+I)=A
  70 LET A(4+I)=-A
  80 NEXT I
  90 LET H=A
 100 FOR I=A TO 8
 110 PRINT AT 7,3*I+A;I
 115 PRINT AT 5,3*I+A;" "
 120 IF A(I)=A THEN PRINT INK 2;
AT 5,3*I+A;"█"
 130 IF A(I)=-A THEN PRINT INK 4
;AT 5,3*I+A;"█"
 140 NEXT I
 150 PRINT PAPER 4; FLASH 1;AT 1
1,1;"JUGADOR 2"
 160 IF NOT (H-A) THEN PRINT PAP
ER 2; FLASH 1;AT 11,1;"JUGADOR 1
"
 170 PRINT "PRINCIPIO , FIN ?";
BEEP 1,12
 180 INPUT D
 190 INPUT F
 195 IF A(D)<>H OR A(F)<>NOT A T
HEN GO TO 150
 200 LET A(D)=NOT A
 210 LET A(F)=A*H
 220 LET H=-H
 230 GO TO 100
```

```
█     █ █   █ █ █

1  2  3  4  5  6  7  8

 JUGADOR 1
PRINCIPIO , FIN ?
```

Letras

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: fácil.
Categoría del juego: de reflexión (para dos jugadores).
Dificultad del juego: mediana.

EL JUEGO: La regla está inspirada en un juego muy conocido en Francia, que se practica especialmente en una emisión de la televisión (la cifras y las letras).

El ZX escoge al azar un cierto número de símbolos (vocales y consonantes) del alfabeto latino, y cada uno de los jugadores deberá indicar la longitud de la palabra que haya logrado obtener (¡el aparato no comprueba si la palabra es correcta!).

La puntuación total de cada jugador se visualiza después.

EL PROGRAMA: El bucle I (2Ø-4Ø) escoge del alfabeto entre 7 y 12 letras en función de la variable aleatoria L, tomada en la línea 1Ø.

Para evitar que hayan demasiadas consonantes se utiliza una astucia en la línea 3Ø, que consiste en volver a sacar al azar información de la tabla A$, que contiene las vocales (pueden obtenerse más o menos utilizando otros RND).

El bucle 5Ø-1ØØ pide a cada jugador la longitud de su palabra.

Las líneas 11Ø y 12Ø comparan los resultados y totalizan la puntuación de cada jugador, que se visualiza por la línea 13Ø.

PARA JUGAR: Se pone en marcha la ejecución del juego con RUN.

En este momento se visualizan las letras, y después la pregunta JR 1. LONG, a la que contesta el primer jugador con la longitud de la palabra que ha obtenido (la corrección de la palabra deberá, sin embargo, estar controlada por el otro jugador); el jugador 2 hará lo mismo, y la puntuación de ambos jugadores será visualizada.

Para continuar la partida bastará con pulsar cualquier tecla (excepto BREAK, que permitirá acabar el juego).

POSIBLES EXTENSIONES: Podrá introducirse una contemporización antes de pedir las respuestas a los jugadores.

También podrá introducirse no sólo la longitud de la palabra sino también la misma palabra; el adversario deberá entonces responder a la pregunta ¿CORRECTA O NO?

Puede ampliarse el juego a más de dos jugadores. ¿Y por qué no intentar definir un juego contra el ZX, al cual se le habría

introducido antes una lista de palabras bastante larga, de la que visualizaría las letras en desorden, con algunas letras de complemento escogidas al azar?

LETRAS

PROGRAMA ZX 81

```
   5 LET S=NOT PI
   7 LET J=S
  10 LET L=INT (6*RND+7)
  15 LET A$="AEIOUE"
  20 FOR I=SGN PI TO L
  30 PRINT AT 2,NOT PI;"LETRAS :
";AT 2,7+I+I;CHR$ INT (26*RND+38
);AT 2,7+I+I;A$(INT (6*RND+SGN P
I)) AND RND<.4
  40 NEXT I
  50 LET I=SGN PI
  60 PRINT AT 5,NOT PI;"JUGADOR
";I;"  LONGITUD ?"
  70 INPUT L/
  80 IF I=SGN PI THEN LET P=L
  90 LET I=I+SGN PI
 100 IF I<INT PI THEN GO TO 60
 110 LET S=S+(P AND P>L)
 120 LET J=J+(L AND L>P)
 130 PRINT "JUGADOR 1:";S;"    JU
GADOR 2:";J
 140 PAUSE 4E4
 150 CLS
 160 GO TO 10


LETRAS : V I D O O H Q X N O H

JUGADOR 2  LONGITUD ?
```

PROGRAMA ZX SPECTRUM

```
   5 LET S=0
   7 LET J=5
  10 LET L=INT (6*RND+7)
  15 LET A$="AEIOUE"
  20 FOR I=1 TO L
  30 PRINT AT 2,0;"LETRAS : ";AT
2,7+2*I;CHR$ INT (26*RND+65);AT
2,7+2*I;A$(INT (6*RND+1)) AND RN
D<.4
  40 NEXT I
  50 LET I=1
  60 PRINT AT 5,0;"JUGADOR ";I;"
LONGITUD ?"
  70 INPUT L
  80 IF I=1 THEN LET P=L
  90 LET I=I+1
 100 IF I<3 THEN GO TO 60
 110 LET S=S+(P AND P>L)
 120 LET J=J+(L AND L>P)
 130 PRINT "JUGADOR 1:";S;"    JU
GADOR 2:";J: BEEP 1,12
 140 PAUSE 4E4
 150 CLS
 160 GO TO 10


LETRAS : E E E W E L A M O S A

JUGADOR 2  LONGITUD ?
JUGADOR 1:5   JUGADOR 2:0
```

Espía

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: fácil.

Categoría del juego: juego de reflexión.

Dificultad del juego: fácil.

EL JUEGO: El programa puede servir, simultáneamente, de codificador y de descifrador de mensajes secretos, siguiendo un método muy fácil de emplear en este tipo de aparato.

Un pequeño generador aleatorio, situado al final del programa, es utilizado para modificar de manera arbitraria cada letra del mensaje; deberá tenerse cuidado de utilizar siempre el mismo número (núcleo), para empezar la codificación o el desciframiento de dos mensajes idénticos.

EL PROGRAMA: El núcleo inicial (que será un entero cualquiera) se introduce en la línea 2Ø, y el mensaje en la línea 4Ø.

Las líneas 5Ø y 6Ø permiten escoger entre las opciones de codificación o de desciframiento.

El bucle I (9Ø-14Ø) se utiliza para efectuar la codificación o el desciframiento.

La línea 11Ø llama al generador aleatorio.

La línea 12Ø realiza la transposición de las letras.

Las líneas 123 y 125 verifican que se esté dentro de los límites autorizados.

Los resultados se imprimirán por la línea 15Ø.

Las líneas 1ØØØ a 1Ø3Ø contienen el generador.

PARA JUGAR: Poner en marcha el programa con RUN.

Introducir a continuación un entero N, que servirá de núcleo para el generador (habrá que guardarlo para el desciframiento).

Después teclear el mensaje seguido de NEWLINE.

Escoger la opción C para CODIFICAR y la opción D para DESCIFRAR, y después pulsar NEWLINE.

El mensaje codificado o descifrado, según el caso, será entonces visualizado.

POSIBLES EXTENSIONES: La codificación podrá ampliarse a otros signos, además de las letras del alfabeto; utilizar, eventualmente, varios generadores; introducir en el mensaje una codificación del o de los números que hayan servido de generadores, etc.

ESPÍA

PROGRAMA ZX 81

```
  10 PRINT "CODIGO :";
  20 INPUT N
  22 PRINT N
  25 LET A=SGN PI/N
  30 PRINT "MENSAJE :";
  40 INPUT A$
  42 PRINT A$
  50 PRINT "(C)ODIFICAR O (D)ESC
IFRAR :";
  60 INPUT R$
  62 PRINT R$
  90 FOR I=SGN PI TO LEN A$
  95 LET T=CODE A$(I)
 110 GO SUB 1000
 120 LET T=T+A*((R$="C")-(R$="D"
))
 123 IF T>63 THEN LET T=T-26
 125 IF T<38 THEN LET T=T+26
 130 LET A$(I)=CHR$ (T)
 140 NEXT I
 150 PRINT "RESULTADO :";A$;W
1000 LET A=(A+PI)**8
1010 LET A=A-INT A
1020 LET A=INT (26*A+SGN PI)
1030 RETURN
```

```
NUMERO DE LETRAS (DE 1 A 12) 6
PRINCIPIO :AEFDBC
FINAL     :ABCDEF

RESULTADO 2 :BCDFEA

 PERMUTACION :2


RESULTADO :BCDFEA


CODIGO :1234
MENSAJE :FELIZ CUMPLEANOS
(C)ODIFICAR O (D)ESCIFRAR :C
RESULTADO :JWMRN;LINYZFJBPB
CODIGO :1234
MENSAJE :JWMRN;LINYZFJBPB
(C)ODIFICAR O (D)ESCIFRAR :D
RESULTADO :FELIZTCUMPLEANOS
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "CODIGO :";
  20 INPUT N: PRINT N
  25 LET A=1/N
  30 PRINT "MENSAJE :";
  40 INPUT A$: PRINT INK 5;A$
  50 PRINT "(C)ODIFICAR O (D)ESC
IFRAR :";
  60 INPUT R$: PRINT R$
  90 FOR I=1 TO LEN A$
  95 LET T=CODE A$(I)
 110 GO SUB 1000
 120 LET T=T+A*((R$="C")-(R$="D"
))
 123 IF T>90 THEN LET T=T-26
 125 IF T<65 THEN LET T=T+26
 130 LET A$(I)=CHR$ (T)
 140 NEXT I
 150 PRINT "RESULTADO :"; INK 2;
A$: BEEP 1,10: BEEP 1,4: STOP
1000 LET A=(A+PI)↑8
1010 LET A=A-INT A
1020 LET A=INT (26*A+1)
1030 RETURN
```

```
CODIGO :1234
MENSAJE :FELIZ CUMPLEANOS
(C)ODIFICAR O (D)ESCIFRAR :C
RESULTADO :JWMRN;LINYZFJBPB
CODIGO :1234
MENSAJE :JWMRN;LINYZFJBPB
(C)ODIFICAR O (D)ESCIFRAR :D
RESULTADO :FELIZTCUMPLEANOS
```

```
CODIGO :1234
MENSAJE :HASTA LA VISTA
(C)ODIFICAR O (D)ESCIFRAR :C
RESULTADO :LSTCO;UO;EWTCO
```

Enganchar los vagones

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: solitario.
Dificultad del juego: fácil.

EL JUEGO: Se trata de enganchar lo más rápidamente posible la locomotora a su vagón, que aparecen al principio de la partida en dos sitios más o menos lejanos de la pantalla. Para ello deberá desplazarse la locomotora, indicando la serie de movimientos que tienen que efectuarse (bajo forma codificada : A para una casilla hacia arriba ; B para una casilla hacia abajo ; D para una casilla hacia la derecha, e I para una casilla hacia la izquierda). Se tecleará por ejemplo: BBIDAAA.

EL PROGRAMA: Las líneas 5 y 1Ø dibujan la locomotora, la línea 2Ø el vagón y las ruedas están en la línea 3Ø.

La posición inicial de la locomotora está definida por las líneas 4Ø y 5Ø, y la del vagón por las líneas 6Ø y 7Ø.

La línea 8Ø evita que inicialmente estén demasiado cerca uno del otro.

La línea 1ØØ realiza la visualización del vagón, y la línea 11Ø la de la locomotora.

La comprobación del final está en la línea 115.

Los movimientos que hay que efectuar se introducen en la línea 13Ø, en una variable alfanumérica R$, y son descifrados en un bucle I (14Ø-17Ø), para ejecutarse en las líneas 18Ø-19Ø.

PARA JUGAR: RUN permite visualizar las posiciones de salida de la locomotora y del vagón.

Se teclea la serie de desplazamientos que hay que efectuar bajo la forma DDDAAABAII, seguido de NEWLINE.

La locomotora se visualiza en su nueva posición, y se vuelve a empezar hasta que se enganche el vagón.

Entonces se obtendrá la visualización del resultado.

POSIBLES EXTENSIONES: Será interesante prever la pérdida de la partida por accidente, si en el curso del desplazamiento la locomotora colisiona con el vagón.

También podrán preverse varios vagones y un orden de enganche, jugar varias personas con varias locomotoras, dibujar las vías del tren, introducir obstáculos que tengan que rodearse obligatoriamente por un cierto lado, etc.

ENGANCHAR LOS VAGONES

PROGRAMA ZX 81

```
   6 LET D$=""
  10 LET A$=""
  20 LET B$=""
  30 LET C$="O O"
  35 LET N=NOT PI
  40 LET LY=INT (CODE ":"*RND)
  50 LET LX=INT (CODE ")"*RND)+S
GN PI
  60 LET WY=INT (CODE ":"*RND)
  70 LET WX=INT (CODE ")"*RND)+S
GN PI
  80 IF ABS (LX-WX)<4 THEN GO TO
 40
  90 CLS
 100 PRINT AT WX,WY;B$;AT WX+SGN
 PI,WY;C$
 110 PRINT AT LX-SGN PI,LY;D$;AT
 LX,LY;A$;AT LX+SGN PI,LY+SGN PI
;C$
 115 IF LX=WX AND WY=LY-INT PI T
HEN GO TO CODE "ABS "
 120 PRINT AT CODE "=",SGN PI;"C
AMINO A SEGUIR (DIIA...)"
 130 INPUT R$
 132 LET X=NOT PI
 134 LET Y=X
 136 LET N=N+SGN PI
 140 FOR I=SGN PI TO LEN R$
 150 LET X=X+(R$(I)="B")-(R$(I)=
"A")
 160 LET Y=Y+(R$(I)="D")-(R$(I)=
"I")
 170 NEXT I
 180 LET LX=LX+X
 190 LET LY=LY+Y
 200 GO TO 90
 210 PRINT AT CODE "=",SGN PI;"B
RAVO ; ENCONTRADO EN ";N;" INTEN
TOS"
```



CAMINO A SEGUIR (DIIA...)

PROGRAMA ZX SPECTRUM

```
   6 LET D$=""
  10 LET A$=""
  20 LET B$=""
  30 LET C$="O O"
  35 LET N=0
  40 LET LY=INT (25*RND)
  50 LET LX=INT (17*RND)+1
  60 LET WY=INT (25*RND)
  70 LET WX=INT (17*RND)+1
  80 IF ABS (LX-WX)<4 THEN GO TO
 40
  90 BORDER 6: PAPER 4: CLS
 100 PRINT INK 2;AT WX,WY;B$; IN
K 0;AT WX+1,WY;C$
 110 PRINT INK 3;AT LX-1,LY;D$;A
T LX,LY;A$; INK 0;AT LX+1,LY+1;C
$
 115 IF LX=WX AND WY=LY-3 THEN G
O TO 210
 120 PRINT AT 21,1;"CAMINO A SEG
UIR (DIIA...)"
 130 INPUT R$
 132 LET X=0
 134 LET Y=X
 136 LET N=N+1
 140 FOR I=1 TO LEN R$
 150 LET X=X+(R$(I)="B" AND LX<2
0)-(R$(I)="A" AND LX>0): BEEP 0.
3,18
 160 LET Y=Y+(R$(I)="D" AND LY<2
8)-(R$(I)="I" AND LY>0)
 170 NEXT I
 180 LET LX=LX+X
 190 LET LY=LY+Y
 200 GO TO 90
 210 PRINT FLASH 1;AT 21,1;"BRAV
O ; ENCONTRADO EN "; BRIGHT 1;N;
 BRIGHT 0;" INTENTOS": BEEP 1,12
```



CAMINO A SEGUIR (DIIA...)

Tirada de un dado trucado

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: mediana.
Categoría del juego: de reflexión.
Dificultad del juego: mediana.

EL JUEGO: Toma la idea de un programa anterior a éste (DADO TRUCADO), en una aplicación estadística: se elabora un dado voluntariamente trucado, que va a tirarse un cierto número de veces, y se prepara una tabla con los resultados.

EL PROGRAMA: Las probabilidades que se desean en cada tirada se introducen por un bucle I (2Ø-8Ø), con todas las verificaciones necesarias de validez.

El número de tiradas que se desean se introducen en la línea 96.

El bucle J (1Ø5-16Ø) contabiliza las tiradas.

El bucle I (12Ø-14Ø) garantiza la conformidad de cada tirada, con las probabilidades introducidas al principio del juego.

Los resultados se contabilizan en la línea 145 y se visualizan gracias a la línea 15Ø.

PARA JUGAR: Empezar la ejecución pulsando RUN, después, a la pregunta PROBA. DEL DADO 1, introducir un número comprendido entre Ø y 1.

Hacer lo mismo para los dados 2, 3, etc., sin olvidar que el total de probabilidades introducidas tiene que ser muy próximo a 1.

Después, ante la visualización de la pregunta NÚM. DE TIRADAS, se introducirá un entero suficientemente grande, seguido de NEWLINE.

El tiempo que tarda en aparecer la visualización de los resultados será más largo cuanto mayor sea el número introducido.

Los resultados se visualizarán debajo de los números de las tiradas y aparecerán con caracteres invertidos: así, 32.3, que aparece debajo del 1 invertido, significa que la tirada de 1, a la que se había asignado una probabilidad de 0.3, se ha realizado 32 veces.

POSIBLES EXTENSIONES: Podrá mejorarse la representación de los resultados utilizando, por ejemplo, un histograma o cualquier otro método gráficamente más expresivo.

TIRADA DE UN DADO TRUCADO

PROGRAMA ZX 81

```
  10 LET P=NOT PI
  15 DIM E(6)
  20 FOR I=SGN PI TO 6
  30 PRINT AT INT PI,NOT PI;"PRO
BABILIDAD DEL DADO ";I
  40 INPUT E(I)
  50 LET P=P+E(I)
  60 IF E(I)<NOT PI OR E(I)>SGN
PI OR P>1.1 THEN GO TO CODE "COS
"
  70 PRINT AT NOT PI,5*I-5;"DADO
 ";I;AT SGN PI,5*I-5;E(I)
  80 NEXT I
  90 IF P<.9 THEN RUN
  95 PRINT AT INT PI,NOT PI;"NUM
ERO DE TIRADAS                  "
  96 INPUT P
 100 RAND
 105 FOR J=SGN PI TO P
 110 LET N=INT (RND*6+SGN PI)
 120 FOR I=SGN PI TO 6
 130 IF N=I AND RND>E(I)-INT E(I
) THEN GO TO 110
 140 NEXT I
 145 LET E(N)=E(N)+SGN PI
 150 PRINT AT PI*PI,5*N-5;E(N);A
T 8,5*N-4;CHR$ (156+N)
 160 NEXT J
 170 STOP
 200 PRINT "IMPOSIBLE :0<= P <=1
"
 210 LET P=P-E(I)
 215 PAUSE 50
 220 CLS
 230 GO TO 30
```

```
DADO DADO DADO DADO DADO DADO 6
0.2  0.1  .05  0.15 0.1  0.4

NUMERO DE TIRADAS



 1    2    3    4    5    6

16.2 7.1  7.05 22.1510.1 38.4
```

PROGRAMA ZX SPECTRUM

```
  10 LET P=0
  15 DIM E(6)
  20 FOR I=1 TO 6
  30 PRINT AT 3,0;"PROBABILIDAD
DEL DADO ";I
  40 INPUT E(I)
  50 LET P=P+E(I)
  60 IF E(I)<0 OR E(I)>1 OR P>1.
1 THEN GO TO 200
  70 PRINT INK 3;AT 0,5*I-5;"DAD
O ";I;AT 1,5*I-5;E(I)
  80 NEXT I
  90 IF P<.9 THEN RUN
  95 PRINT INVERSE 1;AT 3,0;"NUM
ERO DE TIRADAS "; INVERSE 0;"
"
  96 INPUT P: PRINT INK 4;AT 3,1
9;P
 100 RANDOMIZE
 105 FOR J=1 TO P
 110 LET N=INT (RND*6+1)
 120 FOR I=1 TO 6
 130 IF N=I AND RND>E(I)-INT E(I
) THEN GO TO 110
 140 NEXT I
 145 LET E(N)=E(N)+1
 150 PRINT AT 10,5*N-5;E(N); INK
2; INVERSE 1;AT 8,5*N-4;N
 160 NEXT J
 170 STOP
 200 PRINT FLASH 1;"IMPOSIBLE :0
<= P <=1"
 210 LET P=P-E(I)
 215 PAUSE 50
 220 CLS
 230 GO TO 30
```

```
DADO DADO DADO DADO DADO DADO 6
.05  0.4  0.3  0.1  .05  0.1

NUMERO DE TIRADAS  100



 ■    ■    ■    ■    ■    ■

5.05 50.4 24.3 8.1  3.05 10.1
```

Orden

REFERENCIAS:

Modalidad: SLOW o FAST.

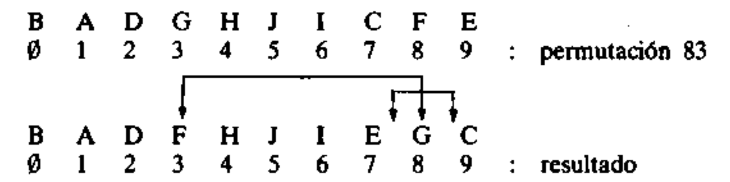
Dificultad de programación: mediana.

Categoría del juego: juego de reflexión.

Dificultad del juego: mediana.

EL JUEGO:

Se visualizan en desorden las 10 primera letras del alfabeto, que deben colocarse en orden, con el mínimo de intentos posibles, por permutaciones sucesivas de dos letras; esto tiene el inconveniente de que hace también permutar las dos letras que rodean la letra que se cambia, siguiendo el mecanismo que se explica abajo:



EL PROGRAMA:

Los bucles intercalados I (1Ø-7Ø) y J (3Ø-5Ø) definen una variable alfanumérica A$, la cual contiene en desorden las 10 primeras letras del alfabeto.

La línea 9Ø visualiza A$, y la línea 1ØØ los números de las letras que se utilizan para la permutación.

Se introduce la permutación en la línea 12Ø.

Las líneas 132 y 135 transforman la variable alfanumérica introducida en una variable numérica.

Las líneas 15Ø a 17Ø realizan la primera inversión.

Las líneas 19Ø a 21Ø efectúan la segunda.

La línea 22Ø contabiliza los intentos.

La línea 23Ø controla los resultados.

PARA JUGAR:

Poner en marcha el programa con RUN.

Se visualiza la palabra inicial, señalando cada letra con un número del Ø al 9.

A la pregunta INVERSION (I/J), se contesta introduciendo un número de dos cifras *xy;* donde *x* es el número de la primera letra e *y* el de la segunda.

Entonces se efectúan las dos inversiones sucesivamente y se vuelve a empezar.

El número de intentos se visualizará al final de la partida.

POSIBLES EXTENSIONES: Puede imaginarse un juego contra el ordenador, el cual evaluaría la puntuación del jugador en función de sus propios resultados. También pueden modificarse las reglas que rigen las permutaciones.

ORDEN

PROGRAMA ZX 81

```
   5 LET A$=""
  10 FOR I=SGN PI TO 10
  20 LET R$=CHR$ (INT (10*RND+38
))
  30 FOR J=SGN PI TO I-SGN PI
  40 IF R$=A$(J) THEN GO TO 20
  50 NEXT J
  60 LET A$=A$+R$
  70 NEXT I
  80 LET S=NOT PI
  90 PRINT AT PI,NOT PI;A$
 100 PRINT "[redacted]"
 110 PRINT AT PI+PI,NOT PI;"INVE
RSION (I/J) ; ";S
 120 INPUT R$
 130 PRINT R$
 132 LET I=CODE R$(SGN PI)-27
 135 LET J=CODE R$(2)-27
 150 LET N$=A$(J)
 160 LET A$(J)=A$(I)
 170 LET A$(I)=N$
 180 IF R$(SGN PI)="0" OR R$(SGN
PI)="9" THEN GO TO 220
 190 LET N$=A$(I-SGN PI)
 200 LET A$(I-SGN PI)=A$(I+SGN P
I)
 210 LET A$(I+SGN PI)=N$
 220 LET S=S+SGN PI
 230 IF A$="ABCDEFGHIJ" THEN PRI
NT "ENCONTRADO EN ";S;" INTENTOS
 :";A$;W
 240 GO TO 90
```

```
ABCDGFHEJI
[redacted]

INVERSION (I/J) ; 4
53
```

PROGRAMA ZX SPECTRUM

```
   5 LET A$=""
  10 FOR I=1 TO 10
  20 LET R$=CHR$ (INT (10*RND+65
))
  30 FOR J=1 TO I-1
  40 IF R$=A$(J) THEN GO TO 20
  50 NEXT J
  60 LET A$=A$+R$
  70 NEXT I
  80 LET S=0
  90 PRINT INVERSE 1; INK 2;AT 3
,0;A$
 100 PRINT PAPER 4;"0123456789"
 110 PRINT AT 6,0;"INVERSION (I/
J) ; INTENTO ";S
 120 INPUT R$
 130 PRINT "I = ";R$(1);" ; J =
";R$(2): BEEP 1,10: BEEP 1,4
 132 LET I=CODE R$(1)-47
 135 LET J=CODE R$(2)-47
 150 LET N$=A$(J)
 160 LET A$(J)=A$(I)
 170 LET A$(I)=N$
 180 IF R$(1)="0" OR R$(1)="9" T
HEN GO TO 220
 190 LET N$=A$(I-1)
 200 LET A$(I-1)=A$(I+1)
 210 LET A$(I+1)=N$
 220 LET S=S+1
 230 IF A$="ABCDEFGHIJ" THEN PRI
NT FLASH 1; PAPER 4;"ENCONTRADO
EN ";S;" INTENTOS :"; FLASH 0; P
APER 5;A$: BEEP 1,10: STOP
 240 GO TO 90
```

```
[redacted]
0123456789

INVERSION (I/J) ; INTENTO 10
I = 5 ; J = 2
```

Carrera de caballos

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: de azar (para varios jugadores).

Dificultad del juego: ninguna.

EL JUEGO: Se introduce un número de caballos, comprendido entre 1 y 10, y la carrera empezará inmediatamente.

Puede haber ex-aequos, ya que se tendrá en cuenta el hecho de que el ZX se ocupa de que los caballos avancen unos después de otros, empezando por el número 1; la clasificación se visualizará a medida que los caballos vayan llegando (teniendo en cuenta los ex-aequo).

EL PROGRAMA: Un primer bucle I (6Ø-15Ø) visualiza los caballos en posición de salida.

Un segundo bucle I (9Ø-15Ø), que utiliza el mismo NEXT I, los hará avanzar; se señalará la posición de cada caballo por una cifra o letra, en una variable alfanumérica A$.

La línea 11Ø hará avanzar los caballos.

La línea 12Ø controla las llegadas.

La línea 14Ø realizará las impresiones.

El subprograma 2ØØ-23Ø avanza el contador de llegadas e imprime los resultados.

PARA JUGAR: Se pone en marcha el programa con RUN.

A la pregunta NÚM. DE CABALLOS :, se responde con un entero del 1 al 10, seguido de NEWLINE. La carrera se celebrará, a continuación, automáticamente.

POSIBLES EXTENSIONES: En un ZX Spectrum se podrán dibujar los caballos con un grafismo más cuidadoso y diferenciándolos por su color.

También podrá imaginarse un sistema de apuestas, que tendrá lugar antes de la salida de cada carrera.

CARRERA DE CABALLOS

PROGRAMA ZX 81

```
  10 LET A$=""
  20 PRINT "NUMERO DE CABALLOS :
";
  30 INPUT N
  35 PRINT N
  40 LET P=NOT PI
  50 LET T=SGN PI
  55 LET A=P
  60 FOR I=T TO N
  70 LET A$=A$+"0"
  80 GO TO 140
  90 FOR I=SGN PI TO N
 100 IF A$(I)="R" THEN GO TO 150
 110 LET A$(I)=CHR$ (CODE A$(I)+
INT (RND*4)+SGN PI)
 120 IF A$(I)>="R" THEN GO SUB 2
00
 140 PRINT AT I+I,CODE A$(I)-28;
"      ";CHR$ (156+I)
 150 NEXT I
 160 LET P=P+T
 170 LET T=NOT PI
 180 IF A<N THEN GO TO 90
 190 STOP
 200 LET A$(I)="R"
 210 PRINT AT I+I,NOT PI;P
 220 LET T=SGN PI
 230 LET A=A+T
 240 RETURN
```

```
NUMERO DE CABALLOS :5
```

PROGRAMA ZX SPECTRUM

```
  10 LET A$=""
  20 PRINT "NUMERO DE CABALLOS :
";
  30 INPUT N
  35 PRINT N
  40 LET P=0
  50 LET T=1
  55 LET A=P
  60 FOR I=T TO N
  70 LET A$=A$+"0"
  80 GO TO 140
  90 FOR I=1 TO N
 100 IF A$(I)="K" THEN GO TO 150
 110 LET A$(I)=CHR$ (CODE A$(I)+
INT (RND*4)+1)
 120 IF A$(I)>="K" THEN GO SUB 2
00
 140 PRINT AT I+I,CODE A$(I)-48;
"      "; INVERSE 1; INK I-7*INT (
I/7);CHR$ (96+I): BEEP 0.2,15: P
AUSE 10
 150 NEXT I
 160 LET P=P+T
 170 LET T=0: PAUSE 20
 180 IF A<N THEN GO TO 90
 190 BEEP 2,0: STOP
 200 LET A$(I)="K"
 210 PRINT AT I+I,0; FLASH 1; PA
PER 5;P
 220 LET T=1
 230 LET A=A+T
 240 RETURN
```

```
NUMERO DE CABALLOS :6
```

El juego de Marienbad

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: mediana.

Categoría del juego: lógica (para dos jugadores).

Dificultad del juego: mediana.

EL JUEGO: Un juego que se ha hecho muy famoso gracias a una película que lo popularizó; no obstante, se presenta aquí en una versión más conforme con sus orígenes, es decir, el juego de NIM.

Al principio de la partida es posible escoger el número de montones que se quiera hacer y el número de cerillas que se quiera poner en cada montón.

Se juega por turnos y pierde el jugador que quita la última cerilla.

EL PROGRAMA: La línea 4Ø dimensiona una variable A(I) que contendrá el número de cerillas de cada montón.

El bucle I (5Ø-56) permite introducir este número.

El bucle I (59-8Ø) imprime los montones.

La línea 1Ø5 asegura el cambio de jugador.

La línea 13Ø, así como la línea 16Ø, controlan si la jugada está autorizada.

La línea 215 controla el final de la partida.

PARA JUGAR: Se pone en marcha el programa con RUN; después se introduce el número de montones (entre 1 y 8), seguido de NEWLINE.

A continuación, se introducirá el número de cerillas en cada montón (pulsar N seguido de NEWLINE en cada introducción). Se obtendrá entonces la visualización de los distintos montones, con el número del jugador concernido (1 o 2), lo mismo que el mensaje NÚM. DE MONTÓN; se responde tecleando el número del montón del que se desea retirar las cerillas, seguido de NEWLINE.

A continuación, se obtiene la pregunta NÚM. A QUITAR, y se responde indicando el número de cerillas que se desea quitar del referido montón.

A continuación será el turno del segundo jugador, y se jugará alternativamente hasta que no haya más cerillas disponibles.

Al final de la partida, se visualizará el número del jugador que ha ganado.

POSIBLES EXTENSIONES: Podrá designarse a cada jugador por su nombre, hacer un grafismo más elaborado, escribir un programa que permita al ZX actuar de pareja, etc.

EL JUEGO DE MARIENBAD

PROGRAMA ZX 81

```
   5 CLS
  10 PRINT "NUMERO DE MONTONES "

  20 INPUT N
  25 PRINT N
  32 LET S=NOT PI
  35 IF N>=8 THEN RUN
  40 DIM A(N)
  50 FOR I=SGN PI TO N
  52 PRINT AT SGN PI,SGN PI;"MON
TON ";I;"; NUMERO DE CERILLAS ?"
  53 INPUT A(I)
  55 LET S=S+A(I)
  56 NEXT I
  57 LET H=SGN PI
  58 CLS
  59 FOR I=SGN PI TO N
  60 PRINT AT 4,4*I+4;"  ";A(I);"
"
  65 PRINT AT 6,SGN PI;"MONTON"
  70 PRINT AT 6,4*I+5;I
  80 NEXT I
 100 PRINT AT 9,4;"JUGADOR ";H
 105 IF H-SGN PI THEN LET H=NOT
PI
 110 PRINT AT 11,SGN PI;"TOMA DE
L MONTON :";
 120 INPUT T
 125 IF T>N THEN GO TO 110
 130 IF A(T)=NOT PI THEN GO TO 1
10
 135 PRINT T
 140 PRINT AT 13,SGN PI;"NUMERO
A QUITAR :    "
 150 INPUT O
 160 IF O>A(T) OR O=NOT PI THEN
GO TO 140
 170 LET A(T)=A(T)-O
 180 LET S=S-O
 210 LET H=H+SGN PI
 215 IF NOT S THEN GO TO 300
 220 GO TO 58
 300 PRINT AT 15,SGN PI;"EL JUGA
DOR ";H;" HA GANADO"
```

```
        2    4    7    9

MONTON  1    2    3    4

   JUGADOR 1

TOMA DEL MONTON :
```

PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 PRINT "NUMERO DE MONTONES "
;
  20 INPUT N: PRINT N
  32 LET S=0
  35 IF N>=8 THEN RUN
  40 DIM A(N)
  50 FOR I=1 TO N
  52 PRINT AT 1,1;"MONTON ";I;";
 NUMERO DE CERILLAS ?"
  53 INPUT A(I)
  55 LET S=S+A(I)
  56 NEXT I
  57 LET H=1
  58 CLS
  59 FOR I=1 TO N
  60 PRINT PAPER 5; BRIGHT 1;AT
4,4*I+4;" ";A(I);" "
  65 PRINT AT 6,1;"MONTON"
  70 PRINT FLASH 1; INK 1;AT 6,4
*I+5;I
  80 NEXT I
 100 PRINT AT 9,4;"JUGADOR ";H
 105 IF H-1 THEN LET H=0
 110 PRINT AT 11,1;"TOMA DEL MON
TON :";: BEEP 1,12
 120 INPUT T: IF T>N THEN GO TO
110
 130 IF A(T)=0 THEN GO TO 110
 135 PRINT T
 140 PRINT AT 13,1;"NUMERO A QUI
TAR :   "
 150 INPUT O: PRINT AT 13,18;O:
PAUSE 22
 160 IF O>A(T) OR O=0 THEN GO TO
140
 170 LET A(T)=A(T)-O
 180 LET S=S-O
 210 LET H=H+1
 215 IF NOT S THEN GO TO 300
 220 GO TO 58
 300 PRINT PAPER 4; INVERSE 1;AT
 15,1;"EL JUGADOR ";H;" HA GANAD
O": BEEP 2,5
```

```
        1    3    5    4

MONTON  1    2    3    4

   JUGADOR 2

TOMA DEL MONTON :3

NUMERO A QUITAR :2
```

Ídem

REFERENCIAS:

Modalidad: SLOW o FAST.

Dificultad de programación: mediana.

Categoría del juego: juego de cálculo mental (para dos jugadores).

Dificultad del juego: según el nivel escogido.

EL JUEGO:

El ZX escoge un número de n cifras (según el nivel de dificultad indicado al principio). Este número (el mismo, al principio, para los dos jugadores) se reducirá en cada jugada una determinada cantidad, igualmente escogida al azar por el ZX al principio del juego; en cada jugada debe escogerse una columna (las decenas, las centenas, etc.) y añadir en esta columna algún entero; el objetivo es obtener, en un momento dado, la visualización de un número que tenga todas sus cifras idénticas.

Cada resultado se visualiza un instante, y a continuación el turno pasará al siguiente jugador, que volverá a partir de su resultado anterior.

EL PROGRAMA:

Una variable numérica N se dimensiona en dos en la línea 12, con el fin de contener los sucesivos resultados de los dos jugadores.

La línea 2Ø permite introducir el nivel de dificultad escogido.

El número inicial se confecciona en la línea 25.

La cantidad que intervendrá en las reducciones se obtiene en la línea 3Ø.

La línea 4Ø comprueba si el resultado no es demasiado pequeño.

La línea 45 visualiza el número del jugador.

El número se visualiza en la línea 5Ø, y la jugada del jugador se introduce en la línea 7Ø (número de 2 cifras).

La línea 8Ø incrementa la columna escogida, y la línea 85 reduce el número de la cantidad escogida al principio.

El resultado se visualizará un instante (líneas 9Ø y 1ØØ). El bucle I (12Ø-135) permite verificar si todas las cifras son idénticas.

La impresión y el final de la partida se efectúan en la línea 14Ø. Las líneas 15Ø y 16Ø se ocupan del cambio de jugador.

PARA JUGAR:

Poner en marcha la ejecución del programa con RUN.

El número inicial se visualiza para el jugador n.° 1, que jugará tecleando *xy*, seguido de NEWLINE; *x* es la cifra que hay que añadir, e *y* es el número de la columna donde debe añadirse.

El resultado (provisional si es demasiado pequeño) se visualizará un

instante; el turno pasa al siguiente jugador, y así sucesivamente. Puede pararse el programa en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Podrá ampliarse el juego a más de dos jugadores, mejorar la elección del número que interviene en la reducción o los distintos controles, dejar visualizados los diferentes resultados, etc.

ÍDEM

PROGRAMA ZX 81

```
  10 LET H=SGN PI
  12 DIM N(H+H)
  15 PRINT "NIVEL (2-7)"
  20 INPUT N
  22 LET U=10
  25 LET N(SGN PI)=INT ((U**N-U*
 (N-SGN PI))*RND+U**(N-SGN PI))
  27 LET N(2)=N(SGN PI)
  30 LET A=INT (2*U**(N-SGN PI)*
RND+U**(N-SGN PI))
  35 CLS
  40 IF N(H)<U**(N-SGN PI) THEN
LET N(H)=U**N-ABS N(H)
  45 PRINT AT INT PI,NOT PI;"JUG
ADOR ";H
  50 PRINT AT PI+PI,NOT PI;"NUME
RO ";TAB 15-N;N(H)
  55 PRINT TAB 8;"[redacted]"
  60 PRINT AT PI+PI,NOT PI;"SU J
UEGO (N/C) ?"
  70 INPUT R$
  80 LET N(H)=N(H)+VAL R$(SGN PI
)*U**(VAL R$(2)-SGN PI)
  85 LET N(H)=N(H)-A
  90 PRINT AT CODE "(",NOT PI;"R
ESULTADO ";N(H)
 100 PAUSE CODE " FAST "
 110 LET R$=STR$ N(H)
 120 FOR I=SGN PI TO LEN R$-SGN
PI
 130 IF R$(I)<>R$(I+SGN PI) THEN
 GO TO 150
 135 NEXT I
 140 PRINT "BRAVO , JUGADOR ";H;
U
 150 LET H=H+SGN PI
 160 IF H>=INT PI THEN LET H=SGN
PI
 170 GO TO CODE "7"
```

```
JUGADOR 2

NUMERO       3076
        [redacted]

SU JUEGO (N/C) ?



RESULTADO 1187
```

PROGRAMA ZX SPECTRUM

```
  10 LET H=1
  12 DIM N(2)
  15 PRINT "NIVEL (2-7)"
  20 INPUT N
  22 LET U=10: LET P=0
  25 LET N(1)=INT ((U↑N-U↑(N-1))
*RND+U↑(N-1))
  27 LET N(2)=N(1)
  30 LET A=INT (2*U↑(N-1)*RND+U↑
(N-1))
  35 CLS : LET P=P+(H=1)
  40 IF N(H)<U↑(N-1) THEN LET N(
H)=U↑N-ABS N(H)
  45 PRINT INVERSE 1; INK H;AT 3
,0;"JUGADOR ";H;AT 3,13;"INTENTO
 ";P
  50 PRINT AT 6,0;"NUMERO ";TAB
15-N; INVERSE 1; INK H;N(H)
  55 PRINT TAB 8; PAPER 5;"76543
21"
  60 PRINT AT 9,0;"SU JUEGO (N/C
) ?"
  70 INPUT R$
  80 LET N(H)=N(H)+VAL R$(1)*U↑(
VAL R$(2)-1)
  85 LET N(H)=N(H)-A
  90 PRINT AT 15,0;"RESULTADO ";
 INVERSE 1; INK H;N(H)
 100 BEEP 1,10: PAUSE 250
 110 LET R$=STR$ N(H)
 120 FOR I=1 TO LEN R$-1
 130 IF R$(I)<>R$(I+1) THEN GO T
O 150
 135 NEXT I
 140 PRINT FLASH 1; PAPER 4;"BRA
VO , JUGADOR ";H: BEEP .3,10: BE
EP .3,18: BEEP .3,8: PRINT "  ";
P;" INTENTOS": STOP
 150 LET H=H+1
 160 IF H>=3 THEN LET H=1
 170 GO TO 35
```

```
[redacted]

NUMERO       [redacted]
        7654321

SU JUEGO (N/C) ?



RESULTADO [redacted]
BRAVO , JUGADOR 1
   1 INTENTOS
```

Mastermind

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: difícil.
Categoría del juego: lógica (para un solo jugador).
Dificultad del juego: difícil.

EL JUEGO: Convertido en un clásico de los juegos sobre microordenadores, este juego desarrolla el razonamiento deductivo; debe encontrarse lo más rápidamente posible un número de n cifras (n deberá estar comprendido entre 1 y 5), disponiendo de información en cada intento. En efecto, el ZX contestará indicando, por una parte, el número de cifras que están en el número propuesto, exactamente en el mismo sitio que el número que se tiene que descubrir, y, por otra parte, el número de cifras que constan en el número que hay que descubrir, que no están exactamente en el lugar apropiado.

EL PROGRAMA: La elección del número de cifras del número que hay que descubrir se efectúa de la línea 1Ø a la línea 3Ø. El número se confecciona en la línea 5Ø (podrá haber cifras repetidas).

La visualización se prepara por la línea 6Ø.

La línea 95 se ocupa del final de la partida.

Un primer bucle I (1ØØ-124) comprueba las cifras que están bien situadas.

Un segundo bucle J (1Ø8-114) comprueba las cifras que están presentes, pero mal colocadas.

Los contadores se encuentran en las líneas 118 y 122.

La línea 128 visualiza el resultado, y la línea 13Ø contabiliza los intentos.

PARA JUGAR: Se pulsa RUN y se escoge la longitud del número deseado (de 1 a 6), seguido de NEWLINE.

Después de la visualización NO. NÚMERO BUENO COLOCADO, se introduce la jugada, seguida de NEWLINE.

Inmediatamente se obtiene la respuesta (en caso de doble cifra, los resultados pueden ser un poco delicados de interpretar).

A continuación se introduce un nuevo intento, y así sucesivamente.

Si la pantalla está llena (mensaje de error 5), habrá que pulsar CONT para borrarla y seguir.

Cuando se encuentra el número, se obtiene la visualización ENCONTRADO EN ... VECES.

Puede salirse del programa en cualquier momento introduciendo una letra, seguida de NEWLINE, en vez del número que se intenta.

POSIBLES EXTENSIONES: Se intentará evitar la presencia de cifras dobles en el número que debe encontrarse; también puede preverse un cambio de página automático, con llamada a los resultados anteriores, efectuarse otros juegos parecidos, con palabras u otros símbolos (color, tamaño), etc.

MASTERMIND

PROGRAMA ZX 81

```
   5 CLS
  10 PRINT "NUMERO DE CIFRAS ";
  20 INPUT N
  25 IF N>6 THEN RUN
  30 PRINT N
  40 LET E=SGN PI
  50 LET X$=STR$ (INT (RND*9**N+
10**(N-SGN PI)))
  60 PRINT AT INT PI,SGN PI;"NO.
  NUMERO   BUENO  COLOCADO"
  70 INPUT E$
  80 LET X=NOT PI
  90 LET Y=X
  95 IF E$=X$ THEN PRINT "ENCONT
RADO ";E$;" EN ";E;" VECES";W
 100 FOR I=SGN PI TO N
 105 IF X$(I)=E$(I) THEN GO TO 1
22
 108 FOR J=SGN PI TC
 112 IF J<>I AND X$(I)=E$(J) THE
N GO TO 118
 114 NEXT J
 116 GO TO 124
 118 LET Y=Y+SGN PI
 120 GO TO 124
 122 LET X=X+SGN PI
 124 NEXT I
 128 PRINT AT 4+E,SGN PI;E;TAB 7
 E$;TAB 18;X;TAB 25;Y
 130 LET E=E+SGN PI
 140 GO TO 60
```

```
NUMERO DE CIFRAS 4

 NO.    NUMERO    BUENO   COLOCADO

 1      1234      0       1
 2      5678      1       2
 3      7419      0       1
 4      3026      0       1
 5      8516      1       2
```

PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 PRINT "NUMERO DE CIFRAS ";
  20 INPUT N
  25 IF N>6 THEN RUN
  30 PRINT N
  40 LET E=1
  50 LET X$=STR$ (INT (RND*9↑N+1
0↑(N-1)))
  60 PRINT FLASH 1; PAPER 4;AT 3
,1;"NO.   NUMERO    BUENO  COLOC
ADO"
  70 INPUT E$
  80 LET X=0
  90 LET Y=X
  95 IF E$=X$ THEN PRINT FLASH 1
; PAPER 2; INK 1;AT 21,0;"ENCONT
RADO ";E$;" EN ";E;" VECES": BEE
P 1,12: STOP
 100 FOR I=1 TO N
 105 IF X$(I)=E$(I) THEN GO TO 1
22
 108 FOR J=1 TO N
 112 IF J<>I AND X$(I)=E$(J) THE
N GO TO 118
 114 NEXT J
 116 GO TO 124
 118 LET Y=Y+1
 120 GO TO 124
 122 LET X=X+1
 124 NEXT I
 128 PRINT INK 1;AT 4+E,1;E;TAB
7;E$;TAB 18;X;TAB 25;Y
 130 LET E=E+1
 140 GO TO 60
```

```
NUMERO DE CIFRAS 4

 NO.    NUMERO    BUENO   COLOCADO

 1      1234      1       0
 2      5678      0       1
 3      9037      2       1
 4      9933      0       0
 5      1077      3       1
 6      1071      2       2
 7      1017      3       1


ENCONTRADO 1007 EN 8 VECES
```

A través

REFERENCIAS:

Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: de reflexión (para dos jugadores).
Dificultad del juego: mediana.

EL JUEGO:

Aparece en pantalla un cuadrado de nueve casillas, con letras sobre el borde vertical y con cifras sobre el borde horizontal.

Uno de los jugadores maneja las cifras y el otro las letras. Pueden efectuarse tres desplazamientos para las cifras y otros tres para las letras.

Estas últimas pueden desplazarse hacia arriba, a la derecha y a la izquierda, y las cifras hacia la derecha, arriba y abajo.

Gana el primero que consigue sacar todos sus símbolos (las letras salen hacia arriba y las cifras hacia la derecha).

No existe la captura y no se autorizan los saltos; tampoco puede utilizarse una casilla que ya esté ocupada.



EL PROGRAMA:

Las posiciones de las cifras y de las letras se graban en una tabla N (4,4); las líneas 5 a 8Ø definen los valores de salida.

Las líneas 11Ø a 15Ø dibujan el campo de juego, y la línea 16Ø visualiza al jugador que tiene el turno (bajo la forma simplificada de 1 para las cifras y A para las letras).

La línea 19Ø controla la jugada, lo mismo que los bucles I (21Ø-226) y J (22Ø-224).

El juego propiamente dicho se controla por las líneas 229 a 4ØØ, y el cambio de jugador se asegura por la línea 41Ø.

PARA JUGAR:

Después de pulsar RUN, se visualiza el campo de juego y el ordenador espera la jugada del jugador designado por 1 o por A.

Primero se introduce la letra o la cifra, que representan el peón que ha de desplazarse, seguido de una indicación de dirección bajo la forma N (para el norte), S (para el sur), E (para el este) y O (para el oeste).

Puede también pulsarse 2E, seguido de NEWLINE. Si la jugada no es válida no se ejecutará y el código del jugador quedará visualizado; si está autorizada, se efectuará el desplazamiento y se obtendrá la visualización correspondiente al segundo jugador. El primero que consiga hacer desaparecer sus tres piezas habrá ganado.

POSIBLES EXTENSIONES: Puede ampliarse el juego con más cifras y letras, así como mejorar los controles. Puede también definirse una estrategia para el ZX.

A TRAVÉS

PROGRAMA ZX 81

```
   5 LET K=SGN PI
   7 LET F=K+K
  10 LET S=F+F
  30 DIM N(S,S)
  40 FOR I=K TO S
  50 LET N(I,K)=CODE "0"+I
  60 LET N(S,I)=CODE "8"+I
  70 NEXT I
  80 LET N(S,K)=NOT PI
 100 CLS
 110 FOR J=SGN PI TO S
 120 FOR I=SGN PI TO S
 130 PRINT AT F*I,F*J;"." AND N(
I,J)=NOT PI;CHR$ N(I,J) AND N(I,
J)<>NOT PI
 140 NEXT I
 150 NEXT J
 160 PRINT AT S*5,SGN PI;"JR ";C
HR$ (CODE "0"+K)
 180 INPUT R$
 190 IF (K<S AND R$(F)="O") OR (
K>S AND R$(F)="S") THEN GOTO COD
E "■"
 210 FOR I=SGN PI TO S
 220 FOR J=SGN PI TO S
 222 IF N(I,J)=CODE R$(SGN PI) T
HEN GOTO CODE " FAST "
 224 NEXT J
 226 NEXT I
 227 GOTO CODE "■"
 229 LET A=I
 230 LET B=J
 320 LET I=I+(R$(F)="S")-(R$(F)=
"N")
 330 LET J=J+(R$(F)="E")-(R$(F)=
"O")
 340 IF (NOT I AND K>S) OR (J>S
AND K<S) THEN GOTO CODE "COS "*F

 350 IF N(I,J)<>NOT PI THEN GOTO
 CODE "■"
 360 LET N(I,J)=CODE R$(SGN PI)
 400 LET N(A,B)=NOT PI
 410 LET K=ABS (CODE "$"-F-K)
 420 GOTO CODE "COS "/F
```

```
      1 . . .

      . 2 B .

      . 3 . C

      . A . .



      JR A
```

PROGRAMA ZX SPECTRUM

```
   5 LET K=1
   7 LET F=2
  10 LET S=4
  30 DIM N(S,S)
  40 FOR I=K TO S
  50 LET N(I,K)=CODE "0"+I
  60 LET N(S,I)=CODE "?"+I
  70 NEXT I
  80 LET N(S,K)=0
 100 BORDER 1: CLS
 110 FOR J=1 TO S
 120 FOR I=1 TO S
 130 PRINT AT F*I,F*J;"." AND N(
I,J)=0; PAPER (2 AND N(I,J)<60)+
(4 AND N(I,J)>60);CHR$ N(I,J) AN
D N(I,J)<>0
 140 NEXT I
 150 NEXT J
 160 PRINT AT S*5,1;"JUGADOR QUE
 TIENE LAS ";"CIFRAS" AND K;"LET
RAS" AND NOT K
 180 INPUT R$
 190 IF (K AND R$(F)="O") OR (NO
T K AND R$(F)="S") THEN GO TO 16
0
 210 FOR I=1 TO S
 220 FOR J=1 TO S
 222 IF N(I,J)=CODE R$(1) THEN G
O TO 229
 224 NEXT J
 226 NEXT I
 227 GO TO 160
 229 LET A=I
 230 LET B=J
 320 LET I=I+(R$(F)="S")-(R$(F)=
"N")
 330 LET J=J+(R$(F)="E")-(R$(F)=
"O")
 340 IF (NOT I AND NOT K) OR (J>
S AND K) THEN GO TO 400
 350 IF N(I,J)<>0 THEN GO TO 160
 360 LET N(I,J)=CODE R$(1)
 400 LET N(A,B)=0
 410 LET K=ABS (K-1): BEEP 1,20*
K-10
 420 GO TO 100
```

```
      . 1 . .

      . . 2 C

      3 . B .

      . A . .



      JUGADOR QUE TIENE LAS CIFRAS
```

Reinas sobre el tablero

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: difícil.
Categoría del juego: de reflexión.
Dificultad del juego: difícil.

EL JUEGO: El programa utiliza un algoritmo muy logrado para resolver un problema que ha dado mucho que hacer a los investigadores, incluso a algunos tan célebres como el matemático GAUSS, que no pudo descubrir todas las soluciones. Se trata de encontrar la forma de colocar las reinas en un tablero de tal forma que ninguna pueda comer a otra.

El programa retenido es muy flexible, ya que tiene la posibilidad de escoger el tamaño del tablero; los resultados se visualizarán a la vez en forma gráfica y en forma numérica.

EL PROGRAMA: La línea 11Ø permite escoger el tamaño del tablero.

Una variable D se dimensiona en consecuencia en la línea 2Ø.

Un bucle principal de exploración I (3Ø-8Ø) contiene un subbucle J (5Ø-7Ø), que realiza el control de no alineamiento (línea 6Ø).

Si durante la exploración se encuentra un alineamiento, se vuelve hacia atrás utilizando las líneas 17Ø y 21Ø y vuelve a empezarse.

El bucle F (1ØØ-14Ø) asegura la impresión del tablero, y el bucle T (115-125), que podrá eventualmente suprimirse, asegura la impresión del resultado bajo la forma de un código numérico correspondiente a la posición de las piezas en cada columna.

La línea 16Ø permite una parada, después de la visualización de cada una de las soluciones.

PARA JUGAR: Se pone en marcha el programa con RUN; después se teclea la dimensión del tablero que se desea y se pulsa NEWLINE.

Basta con esperar pacientemente (cuanto mayor sea el tablero, más larga será la espera).

Se visualizará la primera solución, el programa se parará y habrá que pulsar cualquier tecla (menos BREAK) para que vuelva a ponerse en marcha. Para terminar antes de explorar todas las soluciones deberá pulsarse BREAK.

POSIBLES EXTENSIONES: Pueden encontrarse algoritmos para problemas análogos (posición de las torres, desplazamiento de un caballo, etc.).

REINAS SOBRE EL TABLERO

PROGRAMA ZX 81

```
  10 INPUT S
  20 DIM D(S)
  30 FOR I=SGN PI TO S
  40 LET D(I)=SGN PI
  50 FOR J=SGN PI TO I-SGN PI
  60 IF D(I)=D(J) OR ABS (D(I)-D
(J))=I-J THEN GOTO CODE "█"
  70 NEXT J
  80 NEXT I
  90 CLS
 100 FOR F=SGN PI TO S
 110 PRINT AT NOT PI,F+F;F;AT F+
F,NOT PI;F
 115 FOR T=SGN PI TO S
 120 PRINT AT F+F,T+T;"."
 125 NEXT T
 130 PRINT AT F+F,D(F)+D(F);"█"
 135 PRINT AT CODE "=",F;D(F)
 140 NEXT F
 160 PAUSE 4E4
 170 LET I=I-SGN PI
 180 LET D(I)=D(I)+SGN PI
 190 IF D(I)<=S THEN GOTO CODE "
M"
 200 LET I=I-SGN PI
 210 IF I THEN GOTO CODE "█"
```

```
  1 2 3 4 5 6
1 . █ . . . .
2 . . . █ . .
3 . . . . . █
4 █ . . . . .
5 . . █ . . .
6 . . . . █ .



246135
```

PROGRAMA ZX SPECTRUM

```
  10 INPUT S
  20 DIM D(S)
  30 FOR I=1 TO S
  40 LET D(I)=1
  50 FOR J=1 TO I-1
  60 IF D(I)=D(J) OR ABS (D(I)-D
(J))=I-J THEN GO TO 180
  70 NEXT J
  80 NEXT I
  90 CLS
 100 FOR F=1 TO S
 110 PRINT AT 0,F+F;F;AT F+F,0;F
 115 FOR T=1 TO S
 120 PRINT INK 4;AT F+F,T+T;"*"
 125 NEXT T
 130 PRINT PAPER 1;AT F+F,D(F)+D
(F);" "; BEEP .3,15
 135 PRINT INK 2;AT 21,F;D(F)
 140 NEXT F
 160 PAUSE 4E4
 170 LET I=I-1
 180 LET D(I)=D(I)+1
 190 IF D(I)<=S THEN GO TO 50
 200 LET I=I-1
 210 IF I THEN GO TO 180
```

```
  1 2 3 4 5 6 7 8
1 *   * * * * * *
2 * * *   * * * *
3 * * * * *   * *
4 * * * * * * *
5 * *   * * * * *
6   * * * * * * *
7 * * * * * *   *
8 * * * *   * * *


24683175
```

El juego de la vida

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: difícil.

Categoría del juego: juego solitario (dos versiones).

Dificultad del juego: fácil.

EL JUEGO: La versión que aquí se presenta respeta las reglas establecidas en su origen por el matemático J. H. CONWAY.

Al principio del juego, se define una cierta población, ya sea por el ZX (versión completamente automática) o por el jugador (versión manual).

Esta población irá evolucionando, respetando ciertas reglas muy elementales de nacimiento, de vida y de muerte. La evolución proseguirá hasta que se decida parar el programa.

EL PROGRAMA: Se utilizan dos tablas (B y C) en las líneas 3Ø y 35 para grabar el estado de la población.

Los bucles F (21Ø-25Ø) y G (22Ø-24Ø) se utilizan al principio del juego para definir la población inicial (líneas 222 y 227).

El cálculo de la nueva población se realizará después con la ayuda de un bucle cuádruple: 6Ø-195, 11Ø-19Ø, 12Ø-16Ø, 13Ø-15Ø.

La línea 135 realiza la gestión del nuevo estado siguiendo las reglas.

La visualización de este nuevo estado se obtiene volviendo a utilizar el doble bucle de visualización F (21Ø-25Ø) y G (22Ø-24Ø).

La línea 27Ø permite volver a empezar.

PARA JUGAR: En la versión automática, basta poner en marcha el programa con RUN y mirar.

Se pulsará BREAK para parar.

En la versión manual, la configuración inicial se introducirá, después de la puesta en marcha del programa con RUN, tecleando Ø si no se quiere poner nada en la línea ni en la columna indicadas y 1 si se desea poner una célula.

Después, la evolución proseguirá de manera automática.

POSIBLES EXTENSIONES: Si se dispone de una memoria más importante, será fácil aumentar las dimensiones de las tablas B y C volviendo a definir la variable A.

También podrá introducirse color, mejorar la precisión del dibujo reemplazando los PRINT por los PLOT, etc.

EL JUEGO DE LA VIDA (AUTOMÁTICO)

PROGRAMA ZX 81

```
  10 LET A=INT PI+SGN PI
  20 LET B=SGN PI
  30 DIM B(A,A)
  35 DIM C(A,A)
  40 LET I=NOT PI
  50 GO TO CODE "COS "
  60 FOR F=B TO A
 110 FOR G=B TO A
 115 LET S=NOT PI
 120 FOR I=-B TO B
 130 FOR J=-B TO B
 135 IF F+J=NOT PI OR G+I=NOT PI
 OR F+J=A+B OR G+I=A+B THEN GO T
O CODE "█"
 140 LET S=S+B(F+J,G+I)
 150 NEXT J
 160 NEXT I
 170 LET S=S+S-B(F,G)
 175 LET C(F,G)=NOT PI
 180 IF S>=A+B AND S<=A+A-B THEN
 LET C(F,G)=B
 190 NEXT G
 195 NEXT F
 200 CLS
 210 FOR F=B TO A
 220 FOR G=B TO A
 222 IF I THEN GO TO CODE " FAST
"

 225 PRINT AT PI*A,B;F;",";G
 226 PAUSE A* A
 227 LET C(F,G)=VAL INKEY$
 229 PRINT AT F+F,G+G;"*" AND C(
F,G)=B
 230 LET B(F,G)=C(F,G)
 240 NEXT G
 250 NEXT F
 260 PAUSE A* A
 270 GO TO CODE "W"
```

```
* * *

* * *

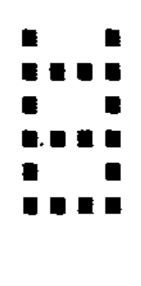
* * *

* * *
```

```
4,4
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "DIMENSION": INPUT A
  20 LET B=1
  30 DIM B(A,A)
  35 DIM C(A,A)
  40 LET I=0
  50 GD TO 200
  60 FOR F=B TO A
 110 FOR G=B TO A
 115 LET S=0
 120 FOR I=-B TO B
 130 FOR J=-B TO B
 135 IF F+J=0 OR G+I=0 OR F+J=A+
B OR G+I=A+B THEN GO TO 150
 140 LET S=S+B(F+J,G+I)
 150 NEXT J
 160 NEXT I
 170 LET S=2*S-B(F,G)
 175 LET C(F,G)=0
 180 IF S>=5 AND S<=7 THEN LET C
(F,G)=B
 190 NEXT G
 195 NEXT F
 200 CLS
 210 FOR F=B TO A
 220 FOR G=B TO A
 222 IF I THEN GO TO 229
 225 PRINT AT 3*A,B;F;",";G
 226 PAUSE A↑A
 227 LET C(F,G)=VAL INKEY$
 229 PRINT INK 2;AT F+F,G+G;"█"
AND C(F,G)=B
 230 LET B(F,G)=C(F,G)
 240 NEXT G
 250 NEXT F
 260 BEEP 1,0
 270 GO TO 60
```



```
6,6
```

EL JUEGO DE LA VIDA (MANUAL)

PROGRAMA ZX 81

```
  10 LET A=INT PI+SGN PI
  20 LET B=SGN PI
  30 DIM B(A,A)
  35 DIM C(A,A)
  40 LET I=NOT PI
  50 GO TO CODE "COS "
  60 FOR F=B TO A
 110 FOR G=B TO A
 115 LET S=NOT PI
 120 FOR I=-B TO B
 130 FOR J=-B TO B
 135 IF F+J=NOT PI OR G+I=NOT PI
 OR F+J=A+B OR G+I=A+B THEN GO T
O CODE "█"
 140 LET S=S+B(F+J,G+I)
 150 NEXT J
 160 NEXT I
 170 LET S=S+S-B(F,G)
 175 LET C(F,G)=NOT PI
 180 IF S>=A+B AND S<=A+A-B THEN
 LET C(F,G)=B
 190 NEXT G
 195 NEXT F
 200 CLS
 210 FOR F=B TO A
 220 FOR G=B TO A
 222 IF I THEN GO TO CODE " FAST
 "
 227 LET C(F,G)=B AND RND<.5
 229 PRINT AT F+F,G+G;"*" AND C(
F,G)=B
 230 LET B(F,G)=C(F,G)
 240 NEXT G
 250 NEXT F
 260 PAUSE A* A
 270 GO TO CODE "W"
```

```
*
  * * *
*   * *
      *
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "DIMENSION": INPUT A
  20 LET B=1
  30 DIM B(A,A)
  35 DIM C(A,A)
  40 LET I=0
  50 GO TO 200
  60 FOR F=B TO A
 110 FOR G=B TO A
 115 LET S=0
 120 FOR I=-B TO B
 130 FOR J=-B TO B
 135 IF F+J=0 OR G+I=0 OR F+J=A+
B OR G+I=A+B THEN GO TO 150
 140 LET S=S+B(F+J,G+I)
 150 NEXT J
 160 NEXT I
 170 LET S=2*S-B(F,G)
 175 LET C(F,G)=0
 180 IF S>=5 AND S<=7 THEN LET C
(F,G)=B
 190 NEXT G
 195 NEXT F
 200 CLS
 210 FOR F=B TO A
 220 FOR G=B TO A
 222 IF I THEN GO TO 229
 227 LET C(F,G)=B AND RND<.5
 229 PRINT INK 2;AT F+F,G+G;"█"
AND C(F,G)=B
 230 LET B(F,G)=C(F,G)
 240 NEXT G
 250 NEXT F
 260 BEEP 1,0
 270 GO TO 60
```

```
█ █   █ █
█ █   █

      █
█ █ █
```

NIVEL 3

N.º	*Nombre del programa*	*Tema y particularidades*
1	**Rombos**	Ayuda a dormirse...
2	**Tablero**	Bases gráficas para creaciones propias.
3	**Pasillos**	Un juego de lógica para descubrir.
4	**Día de la semana**	Se utiliza como recordatorio de los aniversarios.
5	**El ahorcado**	Un juego de letras muy clásico (para dos jugadores).
6	**Bridge**	Para solucionar los problemas del bridge.
7	**Reloj 1**	No hay que tirar la vieja tele: puede transformarse en reloj.
8	**El lobo y los corderos**	Un juego de reflexión para dos jugadores.
9	**Belote**	Si no le gusta el bridge, podrá jugar a la belote.
10	**Un cálculo fastidioso**	Un buen entrenamiento de cálculo (para dos jugadores).
11	**Póquer**	Un inicio para poner a punto el programa.
12	**Hanoï**	Un consejo: ¡empezar poco a poco!
13	**Música**	Revisar las nociones de solfeo gracias al ZX.
14	**Ahorcado al revés**	Esta vez es el ZX el que intenta adivinar la palabra.
15	**Mateo**	¿No sabe dibujar? Deje al ZX.
16	**Máquina de palabras**	¿Es posible que el ZX encuentre palabras que no existan?
17	**Componer**	Para ver, para aprender y para escuchar.
18	**Nicomaque**	Entrenamiento de cálculo mental (uno o varios jugadores).
19	**Permutaciones**	Un juego de reflexión, que no es siempre evidente.
20	**Juego de cartas**	Y para acabar, un poco de prestidigitación, realizada por el ZX.

El manejo de las cadenas y de los gráficos de alta resolución

EL TRATAMIENTO DE LAS CADENAS DE CARACTERES: R$ (⌴TO⌴)

En el nivel 2 se encuentra una primera instrucción que permite extraer el primer elemento de la cadena de caracteres R$; se trata de la instrucción R$(I).

La instrucción que va a manejarse en este nivel da toda su potencia al Basic Sinclair, permitiendo una gran flexibilidad en el manejo de las cadenas de caracteres.

Dada una cadena R$ = "CABALLERO", se tiene la posibilidad de extraer una subcadena que empiece con el carácter I y que termine con el carácter J, escribiendo simplemente R$(I TO J).

De esta manera, R$(1 TO 3) = "CAB"
R$(4 TO 6) = "ALL"

Si se omite I, se toman todos los caracteres de la cadena hasta J (instrucción LEFT$(R$,J) de los Basics normales).

Así, R$(TO 3) = "CAB".

Si se omite J, se toman todos los caracteres de la cadena después de I (instrucción RIGHT$(R$,I) de los Basics normales).

Así, R$(4 TO) = "ALLERO"

El signo + permite encadenar dos cadenas:

R$(TO 3) + R$(4 TO) = "CABALLERO"

Además, es posible sustituir toda una subcadena por otra, con una asignación de cadenas de longitud adecuada.

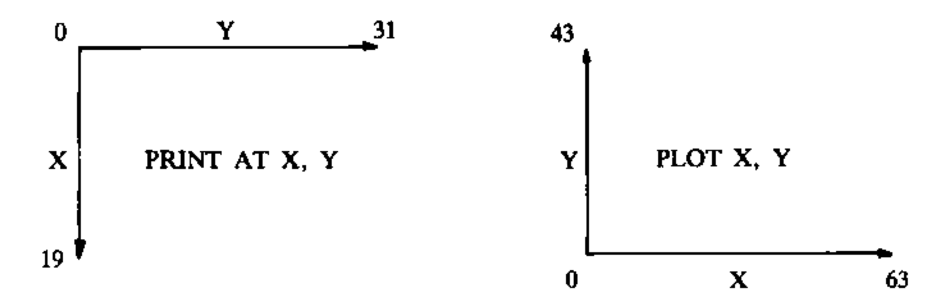
De esta manera, si R$ = "CABALLERO" y si se escribe LET R$(3 TO 5) = "PAT", se habrá obtenido: PRINT R$ = "CAPATLERO".

Las variables I y J pueden, por supuesto, calcularse.

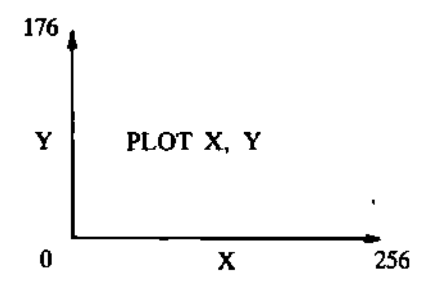
LOS GRÁFICOS DE "ALTA RESOLUCIÓN": LA INSTRUCCIÓN PLOT

El ZX 81 tiene la posibilidad (un poco limitada, a decir verdad) de visualizar o de borrar puntos de la pantalla, con una resolución de 64 × 44 puntos.

De todas maneras, el punto de referencia de los ejes no es el mismo que cuando se utiliza la instrucción PRINT AT.



Un punto visualizado por PLOT 18,23 será borrado, en el ZX 81, por UNPLOT 18,23. En el ZX Spectrum el principio es el mismo, pero la resolución es de 256 × 176.



Un punto visualizado por PLOT 150,79 será borrado, en el ZX Spectrum, por PLOT OVER 1, 150,79.

Rombos

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: fácil.

Categoría del juego: interludio.

Dificultad del juego: ninguna.

EL JUEGO: El programa construirá progresivamente en la pantalla un rombo negro, y después lo hará desaparecer, también progresivamente, lo reconstruirá, y así sucesivamente.

Este juego podría utilizarse para dormirse en caso de que no fuera suficiente con las ovejas.

Igualmente, se propone una versión, ligeramente modificada, que permitirá que aparezcan motivos aleatorios después de la confección de los rombos negros.

EL PROGRAMA: Un primer bucle K (1Ø-6Ø) define las dimensiones de los trazados sucesivos.

El segundo bucle I (2Ø-5Ø) marca las bandas de puntos, utilizando instrucciones PLOT (líneas 25 a 45).

Si la variable S (línea 7Ø) está en 1, se producirá el borrado por la utilización de las instrucciones UNPLOT (líneas 9Ø a 1Ø5).

Para obtener un borrado aleatorio (segundo programa), bastará con modificar la línea 22, que normalmente contiene: 22 IF S THEN GOTO 9Ø, escribiendo 22 IF S THEN GOTO 9Ø + 5 × INT (5 × RND).

Entonces se obtiene un borrado aleatorio.

PARA JUGAR: Se empieza la ejecución del programa con RUN. A partir de este momento, basta con esperar y mirar.

En cualquier momento puede pararse el desarrollo pulsando BREAK.

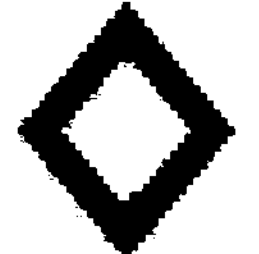
POSIBLES EXTENSIONES: Por supuesto podrá introducirse color y, llegado el caso, sonidos.

A partir de este momento, puede imaginarse toda una serie de programas basados en el mismo principio (caleidoscopio, tapices, etcétera).

ROMBOS

PROGRAMA ZX 81

```
   5 LET S=NOT PI
  10 FOR K=NOT PI TO 15
  20 FOR I=NOT PI TO K
  22 IF S THEN GOTO 90+5*INT (5*
RND)
  25 PLOT 31-K+I,21-I
  30 PLOT 31-K+I,21+I
  35 PLOT 31+K-I,21+I
  40 PLOT 31+K-I,21-I
  50 NEXT I
  60 NEXT K
  70 LET S=NOT S
  80 GOTO 10
  90 UNPLOT 31-K+I,21-I
  95 UNPLOT 31-K+I,21+I
 100 UNPLOT 31+K-I,21+I
 105 UNPLOT 31+K-I,21-I
 110 GOTO 50
```



PROGRAMA ZX SPECTRUM

```
   5 LET S=0: INK 2: PAPER 5: CL
S
  10 FOR K=0 TO 40
  20 FOR I=0 TO K
  22 IF S THEN GO TO 90
  25 PLOT 124-K+I,84-I
  30 PLOT 124-K+I,84+I
  35 PLOT 124+K-I,84+I
  40 PLOT 124+K-I,84-I
  50 NEXT I
  60 NEXT K
  70 LET S=NOT S
  80 GO TO 10
  90 PLOT OVER 1;124-K+I,84-I
  95 PLOT OVER 1;124-K+I,84+I
 100 PLOT OVER 1;124+K-I,84+I
 105 PLOT OVER 1;124+K-I,84-I
 110 GO TO 50
```



ROMBOS (aleatorios)

PROGRAMA ZX 81

```
   5 LET S=NOT PI
  10 FOR K=NOT PI TO 15
  20 FOR I=NOT PI TO K
  22 IF S THEN GOTO 90
  25 PLOT 31-K+I,21-I
  30 PLOT 31-K+I,21+I
  35 PLOT 31+K-I,21+I
  40 PLOT 31+K-I,21-I
  50 NEXT I
  60 NEXT K
  70 LET S=NOT S
  80 GOTO 10
  90 UNPLOT 31-K+I,21-I
  95 UNPLOT 31-K+I,21+I
 100 UNPLOT 31+K-I,21+I
 105 UNPLOT 31+K-I,21-I
 110 GOTO 50
```



PROGRAMA ZX SPECTRUM

```
   5 LET S=0: INK 2: PAPER 5: CL
S
  10 FOR K=0 TO 40
  20 FOR I=0 TO K
  22 IF S THEN GO TO 90+5*INT (5
*RND)
  25 PLOT 124-K+I,84-I
  30 PLOT 124-K+I,84+I
  35 PLOT 124+K-I,84+I
  40 PLOT 124+K-I,84-I
  50 NEXT I
  60 NEXT K
  70 LET S=NOT S
  80 GO TO 10
  90 PLOT OVER 1;124-K+I,84-I
  95 PLOT OVER 1;124-K+I,84+I
 100 PLOT OVER 1;124+K-I,84+I
 105 PLOT OVER 1;124+K-I,84-I
 110 GO TO 50
```



Tablero

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: difícil.
Categoría del juego: grafismo.
Dificultad del juego: variable, según la utilización.

EL JUEGO: El programa que se propone es un complemento del programa AJEDREZ, para utilizarse con una eventual ampliación de la memoria; en efecto, da la posibilidad de realizar una visualización gráfica de buena calidad para los juegos que utilicen un tablero de 8 × 8 casillas (podrá fácilmente ampliarse a 10 × 10 casillas).

Se han previsto unos puntos de referencia, horizontalmente representados por letras (de la A a la H) y verticalmente por números (del 1 al 8).

EL PROGRAMA: Las líneas 1Ø y 2Ø definen dos variables A$ y C$, que contienen los elementos de dibujo.

La línea 3Ø imprime las letras horizontales y la primera línea del dibujo.

El bucle I (4Ø-7Ø) permite la impresión; el motivo que hay que imprimir se escoge en A$ o en C$ por una variable S (línea 5Ø) que toma los valores 1 o 4, según el valor de I; la línea 6Ø efectúa la impresión.

PARA JUGAR: La orden RUN provoca la visualización del tablero en la pantalla.

Para utilizar el programa conviene tener en cuenta los puntos de referencia de las casillas utilizables, que son los centros de los cuadrados negros o grises.

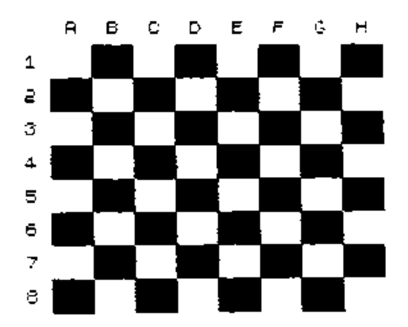
Las casillas se obtienen a lo ancho por una tabulación 3 * I + 1, I variante de 1 a 8, y a lo alto por 2 * I + 1.

POSIBLES EXTENSIONES: Este programa podrá utilizarse en toda clase de juegos que se desarrollen en un tablero (ajedrez, damas inglesas, el lobo y los corderos, etc.).

TABLERO

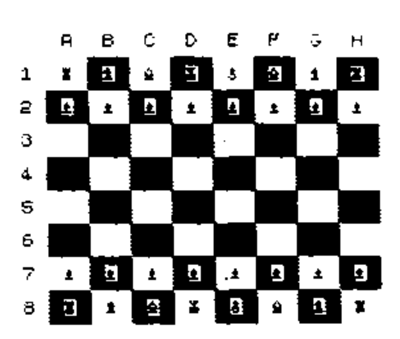
PROGRAMA ZX 81

```
   5 LET S=VAL "2"
  10 LET A$="███"
  20 LET B$="███"
  30 LET C$="███"
  40 FOR I=SGN PI TO VAL "8"
  45 PRINT AT NOT PI,INT PI*I+SG
N PI;CHR$ (CODE "9"+I)
  50 FOR J=SGN PI TO VAL "8"
  60 IF (I+J)/S=INT ((I+J)/S) TH
EN GO TO 80
  70 PRINT AT S*I-SGN PI,INT PI*
J;A$;TAB SGN PI;I;AT S*I,INT PI*
J;B$;AT S*I+SGN PI,INT PI*J;C$
  80 NEXT J
  90 NEXT I
```



PROGRAMA ZX SPECTRUM

```
   2 BORDER 2: CLS
   5 LET S=2
  10 LET A$="███"
  20 LET B$="███"
  30 LET C$="███"
  40 FOR I=1 TO 8
  45 PRINT AT 0,3*I+1;CHR$ (CODE
 "@"+I)
  50 FOR J=1 TO 8
  60 IF (I+J)/S=INT ((I+J)/S) TH
EN GO TO 80
  70 PRINT AT S*I,1;I; INK 1;AT
S*I-1,3*J;A$;AT S*I,3*J,B$;AT S*
I+1,3*J;C$
  80 NEXT J
  90 NEXT I
 100 GO SUB 900
 110 FOR I=1 TO 8
 120 PRINT INK 2;AT 4,3*I+1;CHR$
 159
 130 PRINT INK 4;AT 14,3*I+1;CHR
$ 159
 140 NEXT I
 145 LET H=2: LET P=2
 150 PRINT INK P;AT H,4;CHR$ 163
;AT H,7;CHR$ 154;AT H,10;CHR$ 14
9;AT H,13;CHR$ 160;AT H,16;CHR$
161;AT H,19;CHR$ 149
 160 PRINT INK P;AT H,22;CHR$ 15
4;AT H,25;CHR$ 163
 170 LET H=H+14: LET P=P+2
 180 IF H>20 THEN STOP
 200 GO TO 150
 900 REM DIBUJO DE LAS PIEZAS
 910 FOR i=1 TO 6: READ p$
 920 FOR j=0 TO 7: READ a: POKE
USR p$+j,a
 930 NEXT j
 940 NEXT i
1000 DATA "f",0,16,40,68,108,56,
124,0
1100 DATA "r",0,16,56,16,56,68,5
6,0
1200 DATA "t",0,84,124 56,56,124
,124,0
1300 DATA "q",0,84,40,16,108,124
,124,0
1400 DATA "p",0,0,16,56,56,16,12
4,0
1500 DATA "k",0,16,56,120,24,56,
124,0
1600 RETURN
```



Pasillos

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: de reflexión.
Dificultad del juego: difícil.

EL JUEGO: Este juego recurre a nociones de lógica elemental. El jugador puede presentarse en la entrada del pasillo número 1 o del pasillo número 2. En estas dos entradas se colocan unos torniquetes que basculan al pasar y dejan entrar una vez a la derecha y a la siguiente vez a la izquierda.

Además, un torniquete análogo está dispuesto en el pasillo central; ¿podrá preverse por dónde (salida A o B) se efectuará la próxima salida?

EL PROGRAMA: El bucle I (1Ø-7Ø) dibuja los pasillos (los tres torniquetes no se ven), y las líneas 8Ø y 9Ø imprimen los nombres de las entradas (1 y 2) y de las salidas (A y B).

Las líneas 1ØØ y 11Ø esperan la elección.

El desciframiento del camino lo realiza el ZX utilizando tres variables lógicas I, J y K, que representan el estado de los tres torniquetes (líneas 13Ø a 24Ø).

PARA JUGAR: Después de pulsar RUN, se dibuja el pasillo en la pantalla; los torniquetes no son visibles, y deberá pulsarse 1 o 2 en respuesta a la pregunta del ZX.

En la versión que aquí se retiene, la respuesta se visualiza inmediatamente y el camino que se ha seguido desde el principio del juego, así como las correspondientes salidas.

Esto presenta la ventaja de permitir a varios jugadores ejercer su sagacidad, intentando prever por una entrada dada la respuesta del ZX.

POSIBLES EXTENSIONES: Serán muy numerosas; es muy fácil imaginarse un pasillo más complejo, con otras entradas, salidas y torniquetes, así como una visualización gráfica de los resultados, dibujando, durante un instante, el estado de los torniquetes después de la respuesta del jugador. También puede añadirse una comprobación de las respuestas (juego para varias personas), descontar los puntos de cada jugador, etc.

PASILLOS

PROGRAMA ZX 81

```
  10 FOR I=INT (PI+PI) TO CODE "
3"
  20 PLOT NOT PI,I
  30 PLOT CODE "?",I
  35 IF I<CODE ";" OR I>CODE "-"
 THEN GOTO CODE "U"
  40 PLOT INT PI,I
  50 PLOT CODE "£",I
  60 IF I<CODE "£" OR I>=CODE "=
" THEN PLOT PI+PI+SGN PI,I
  70 NEXT I
  75 LET I=NOT PI
  76 LET J=I
  77 LET K=I
  80 PRINT AT 5,SGN PI;"A";TAB P
I+PI;"B"
  90 PRINT AT CODE "=",SGN PI;"1
";TAB PI+PI;"2"
  95 LET R$=""
  96 LET T$=""
 100 PRINT AT NOT PI,NOT PI;"1 O
2 :";
 110 INPUT C$
 120 LET R$=R$+C$
 130 PRINT C$;TAB CODE "$";R$
 140 IF C$="1" THEN GOTO CODE "A
B5 "
 145 LET J=NOT J
 150 IF J THEN GOTO CODE " FAST
"
 200 LET S$="B"
 205 GOTO CODE " THEN "
 210 LET I=NOT I
 215 IF I THEN GOTO CODE " FAST
"
 220 LET S$="A"
 222 LET T$=T$+S$
 224 PRINT AT INT PI,NOT PI;"SAL
IDA :";S$;"  ";T$
 225 GOTO 100
 229 LET K=NOT K
 235 IF K THEN GOTO CODE "COS "
 240 GOTO CODE ">="
```

```
1 O 2 :1        122211111

SALIDA :A   BABBAAABA
```



PROGRAMA ZX SPECTRUM

```
  10 FOR I=12 TO 140
  20 PLOT 5,I
  30 PLOT 65,I
  35 IF I<45 OR I>100 THEN GO TO
 60
  40 PLOT 20,I
  50 PLOT 50,I
  60 IF I<40 OR I>90 THEN PLOT 3
5,I
  70 NEXT I
  72 PRINT PAPER 5;AT 18,2;"T";A
T 18,6;"T";AT 12,4;"T"
  75 LET I=0
  76 LET J=I
  77 LET K=I
  80 PRINT INK 2; FLASH 1;AT 5,2
;"A";AT 5,6;"B"
  90 PRINT INK 1; FLASH 1;AT 21,
2;"1";AT 21,6;"2"
  95 LET R$=""
  96 LET T$=""
 100 PRINT AT 0,0;"ENTRADA 1,2 :
";
 110 INPUT C$: IF C$="" OR C$>="
3" OR LEN C$<>1 THEN GO TO 100
 120 LET R$=R$+C$
 130 PRINT C$;TAB 12;R$
 140 IF C$="1" THEN GO TO 210
 145 LET J=NOT J
 150 IF J THEN GO TO 229
 200 LET S$="B"
 205 GO TO 222
 210 LET I=NOT I
 215 IF I THEN GO TO 229
 220 LET S$="A"
 222 LET T$=T$+S$
 224 PRINT AT 3,0;"SALIDA :";S$;
TAB 12;T$: BEEP 0.3,12
 226 GO TO 100
 229 LET K=NOT K
 235 IF K THEN GO TO 200
 240 GO TO 220
```

```
ENTRADA 1,2 :1
             1111122222111

SALIDA :A    BAAABABBBAABA

  A   B

    T

  T   T

  1   2
```

Día de la semana

REFERENCIAS:

Modalidad: FAST o SLOW.

Dificultad de programación: mediana.

Categoría del juego: utilitario.

Dificultad del juego: fácil.

EL JUEGO:

El programa permite encontrar, sin ningún tipo de cálculo, el día de la semana que corresponde a una determinada fecha.

La fecha se introduce bajo la forma de tres números, por ejemplo, 11, 3, 1956 para el 11 de marzo de 1956, y el resultado se visualiza claramente (de una manera algo abreviada: MAR por martes, DOM, por domingo...).

EL PROGRAMA:

Las líneas 1Ø a 75 controlan las entradas.

La línea 3Ø se ocupa del final de la partida.

La línea 9Ø permite una corrección de las fechas anteriores a 1582 (puesta en servicio del calendario actual).

Las líneas 1Ø5 a 11Ø permiten la visualización del mes en la respuesta (con 4 letras).

La fórmula que se utiliza aparece en las líneas 13Ø y 135.

Las líneas 14Ø y 15Ø permiten la visualización del día en la respuesta (con 3 letras).

La línea 16Ø imprime el resultado.

El subprograma 2ØØ-22Ø interviene para tener en cuenta los meses de enero y febrero.

PARA JUGAR:

Poner en marcha el programa con RUN, después introducir el día, el mes (entre 1 y 12) y el año (bajo la forma 1845, por ejemplo), cada dato seguido de NEWLINE.

El resultado se visualizará rápidamente, y el programa vuelve a la pregunta DÍA.

Para continuar, basta con pulsar un día, un mes y un año.

Para terminar, se responderá Ø a la pregunta DÍA.

POSIBLES EXTENSIONES:

Pueden introducirse los meses en claro y visualizar el nombre completo del día.

Puede considerarse la visualización de fechas de fiestas particulares (Navidad, Pascua, Domingo de Ramos, Pentecostés).

Puede escribirse un programa que visualizaría los días festivos de un año dado.

DÍA DE LA SEMANA

PROGRAMA ZX 81

```
  10 PRINT AT NOT PI,NOT PI;"DIA
      :";
  20 INPUT J
  30 IF NOT J THEN STOP
  35 PRINT J
  40 PRINT "MES (1-12) ";
  50 INPUT M
  55 PRINT M
  60 PRINT "ANO            ";
  65 INPUT A
  70 PRINT A
  75 LET B=A
  80 LET S=INT (A/100)-INT (A/40
0)
  90 IF A<=VAL "1582" THEN LET S
=VAL "2"
 105 LET M$="ENERFEBRMARZABRIMAY
OJUNIJULIAGOSSEPTOCTUNOVICICI"
 110 LET M$=M$(M*VAL "4"-INT PI
TO M*VAL "4")
 120 IF M<INT PI THEN GO SUB COD
E "COS "
 130 LET S=J-SGN PI+INT (VAL "13
"*(M+SGN PI)/VAL "5")+INT (VAL "
5"*A/VAL "4")-S
 135 LET S=S-VAL "7"*INT (S/VAL
"7")
 140 LET J$="DOMLUNMARMIEJUEVIES
AB"
 150 LET J$=J$(INT PI*S+SGN PI T
O INT PI*S+INT PI)
 160 PRINT AT VAL "7",NOT PI;"EL
 ";J;" ";M$;" ";B;" ERA UN ";J$
 170 GO TO VAL "10"
 200 LET M=M+VAL "12"
 210 LET A=A-SGN PI
 220 RETURN
```

```
DIA         :7
MES (1-12):7
ANO         :1984


EL 7 JULI 1984 ERA UN SAB
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT AT 0,0;"DIA          :";
  20 INPUT J
  30 IF NOT J THEN STOP
  35 PRINT INK 2; FLASH 1;J
  40 PRINT "MES (1-12):";
  50 INPUT M
  55 PRINT INK 2; FLASH 1;M
  60 PRINT "ANO          :";
  65 INPUT A
  70 PRINT INK 2; FLASH 1;A
  75 LET B=A
  80 LET S=INT (A/100)-INT (A/40
0)
  90 IF A<=1582 THEN LET S=2
 105 LET M$="ENERFEBRMARZABRIMAY
OJUNIJULIAGOSSEPTOCTUNOVIDICI"
 110 LET M$=M$(M*4-3 TO M*4)
 120 IF M<3 THEN GO SUB 200
 130 LET S=J-1+INT (13*(M+1)/5)+
INT (5*A/4)-S
 135 LET S=S-7*INT (S/7)
 140 LET J$="DOMINGO  LUNES    M
ARTES    MIERCOLESJUEVES   VIERNE
S  SABADO   "
 150 LET J$=J$(9*S+1 TO 9*S+9)
 160 PRINT INVERSE 1;AT 7,0;"EL
";J;" ";M$;" ";B;" ERA UN "; FLA
SH 1;J$: BEEP 1,12
 170 GO TO 10
 200 LET M=M+12
 210 LET A=A-1
 220 RETURN
```

```
DIA         :14
MES (1-12):7
ANO         :1984


[illegible]
```

El ahorcado

REFERENCIAS:

Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: juego de letras (versión para dos jugadores).
Dificultad del juego: mediana.

EL JUEGO:

Sería, sin duda, bastante difícil encontrar el nombre del inventor de este juego tan de moda, pero no hay ninguna duda de que existe desde tiempo inmemorial; algunas veces también se ha practicado bajo otras formas (por ejemplo, reconstituir un cochinillo tirando los dados).

La aparición de los juegos para microordenadores y la magia de las pantallas de televisión le han asegurado un renacimiento seguro; en la versión retenida, uno de los jugadores debe introducir una palabra de una longitud cualquiera. El ordenador visualiza entonces rayas en el sitio de las letras de esta palabra y pide al segundo jugador que proponga una letra.

Cada respuesta correcta hace aparecer la letra en el sitio exacto de la palabra, y cada respuesta falsa hace progresar el dibujo de la horca.

Para ganar, basta con encontrar la palabra desconocida antes de ser ahorcado.

EL PROGRAMA:

La palabra que hay que adivinar se introduce al principio del programa (líneas 1Ø y 2Ø); una variable B$ en la línea 25 contiene todos los elementos del dibujo.

El bucle I (4Ø-6Ø) confecciona una palabra P$ que contiene las rayas en el sitio de cada una de las letras de la palabra inicial.

El bucle I (84-88) se ocupa del dibujo (más o menos completo, según el valor de P).

Las líneas 9Ø y 92 visualizan el resultado y paran el juego (gracias a una variable W, no definida).

La letra que se prueba se introduce en las líneas 95 y 1ØØ y se comprueba en el bucle I (11Ø-13Ø); el subprograma 2ØØ-22Ø reemplaza cada raya por la letra correspondiente.

PARA JUGAR:

Después de poner en marcha el programa con RUN, uno de los jugadores debe introducir su palabra y pulsar NEWLINE.

Las rayas se visualizan en el sitio de cada una de las letras de la palabra, y la pregunta LETRA aparece en la pantalla; el otro jugador responde proponiendo una letra, seguida de NEWLINE.

Si la letra es correcta, reemplazará a una o a varias rayas; si no, la horca empezará a dibujarse.

Se visualizará la palabra exacta cuando se haya perdido.

POSIBLES EXTENSIONES: **Será interesante confeccionar una lista de palabras, de la que el ZX podrá escoger al azar; también podrá mejorarse el dibujo y modificar el número de intentos que conducen a la horca, en función de la dificultad de la palabra.**

EL AHORCADO

PROGRAMA ZX 81

```
  10 PRINT "SU PALABRA"
  20 INPUT R$
  25 LET B$="█ █ ███████ I █  .  . ██"
  30 LET P$=""
  40 FOR I=SGN PI TO LEN R$
  50 LET P$=P$+"-"
  60 NEXT I
  65 LET P=11
  70 CLS
  80 PRINT AT NOT PI,NOT PI;P$
  84 FOR I=10 TO P STEP -SGN PI
  86 PRINT TAB 14;B$(SGN PI+INT PI*(I-SGN PI) TO I*INT PI)
  88 NEXT I
  90 IF P$=R$ THEN PRINT "BRAVO";W
  92 IF P=SGN PI THEN PRINT "HA PERDIDO : ";R$;W
  95 PRINT AT PI,NOT PI;"SU LETRA"
 100 INPUT S$
 105 LET T=NOT PI
 110 FOR I=SGN PI TO LEN R$
 120 IF R$(I)=S$ THEN GOSUB 200
 130 NEXT I
 170 IF T THEN GOTO 70
 180 LET P=P-SGN PI
 190 GOTO 70
 200 LET T=SGN PI
 210 LET P$(I)=S$
 220 RETURN
```



PROGRAMA ZX SPECTRUM

```
  10 PRINT "SU PALABRA"
  20 INPUT R$
  25 LET B$="█ █ ███AAAAAA A  .  . ██"
  30 LET P$=""
  40 FOR I=1 TO LEN R$
  50 LET P$=P$+"-"
  60 NEXT I
  65 LET P=11
  70 CLS
  80 PRINT AT 0,0;P$
  84 FOR I=10 TO P STEP -1
  86 PRINT TAB 14; INK 2;B$(1+3*(I-1) TO I*3)
  88 NEXT I
  90 IF P$=R$ THEN PRINT FLASH 1; PAPER 4;"BRAVO": BEEP .2,10: BEEP .2,10: BEEP .2,10: BEEP .2,10: STOP
  92 IF P=1 THEN PRINT FLASH 1;"PIERDE : LA PALABRA ERA ";R$: BEEP .2,-10: BEEP .2,-10: BEEP .2,-10: BEEP .2,-10: STOP
  95 PRINT AT 15,0; PAPER 5; INK 1; BRIGHT 1;"SU LETRA"
 100 INPUT S$: BEEP 1,10
 105 LET T=0
 110 FOR I=1 TO LEN R$
 120 IF R$(I)=S$ THEN GO SUB 200
 130 NEXT I
 170 IF T THEN GO TO 70
 180 LET P=P-1
 190 GO TO 70
 200 LET T=1
 210 LET P$(I)=S$
 220 RETURN
```

--L--R-

A

SU LETRA

Bridge

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: difícil.
Categoría del juego: de reflexión.
Dificultad del juego: variable, según el tipo de problema.

EL JUEGO: En este programa no se encontrará una pareja para jugar al bridge, sino, más bien, una simulación de un reparto de cartas; la visualización de este reparto de cartas se realiza juego por juego, sobre tres juegos solamente (a causa del reducido tamaño de la memoria); a partir de aquí, pueden presentarse todo tipo de problemas: declaraciones, carteos, etc.

EL PROGRAMA: Las líneas 1Ø y 15 definen un juego de 52 cartas. La variable B$, en la línea 3Ø, permite dar el nombre abreviado de los colores y también mejorar la designación de algunas cartas (10, valet, dama, rey y as).

El bucle J (4Ø-14Ø) realiza el reparto de tres juegos de 13 cartas (bastará con poner cuatro en lugar de INT PI, en la línea 4Ø, para sacar las 52 cartas).

La línea 5Ø sirve únicamente para volver a colocar la cabeza de la escritura en lo alto de la pantalla.

El bucle I (6Ø-11Ø) realiza el reparto y la impresión. En la línea 66 se escoge la carta entre las 52 disponibles.

La línea 75 calcula el color de la carta que se ha sacado.

La línea 8Ø verifica que esta carta no haya sido sacada anteriormente.

Las líneas 81 y 82 se ocupan de la impresión complementaria, en el caso de que se saque un 10, un valet, una dama, un rey o un as.

La impresión se realiza en la línea 88.

La línea 1ØØ permite eliminar las cartas que ya se han sacado, y la línea 12Ø realiza una parada entre cada visualización.

PARA JUGAR: Se pone en marcha el programa con RUN.

Se visualiza verticalmente una primera serie de 13 cartas. Bastará con pulsar cualquier tecla (menos BREAK) para obtener la segunda serie de 13 cartas. Se procederá de la misma manera para la tercera.

POSIBLES EXTENSIONES: En un material que disponga de una visualización más lograda, como es el caso del ZX Spectrum, podrán representarse las 13 cartas bajo una forma más convencional.

También puede añadirse la posibilidad de hacer declaraciones (y, por ejemplo, compararlas con las del ZX) o introducir posibilidades de carteo, incluso para el ZX.

BRIDGE

PROGRAMA ZX 81

```
  10 LET R$="234567891VDRA"
  15 LET R$=R$+R$+R$+R$
  20 LET V=INT (PI*PI)
  30 LET B$="TRDICOPI0AAES"
  35 LET V=LEN B$
  40 FOR J=SGN PI TO INT PI
  50 PRINT AT NOT PI,V
  60 FOR I=SGN PI TO V
  66 LET X=INT (LEN R$*RND+SGN P
I)
  75 LET S=VAL "2"*INT ((X-SGN P
I)/V)+SGN PI
  80 IF R$(X)="0" THEN GOTO CODE
 "PI"
  81 LET T=X-V*INT (X/V)
  82 LET T=T+V*NOT T
  88 PRINT TAB U*J-U;R$(X)+(B$(T
) AND T>=U);TAB U*J-PI-PI;B$(S T
O S+SGN PI)
 100 LET R$(X)="0"
 110 NEXT I
 120 PAUSE LEN R$*V
 140 NEXT J
```

```
8  TR     4  DI     9  PI
4  TR     3  PI     5  CO
AS DI     6  PI     RE PI
DA TR     RE DI     4  CO
10 CO     2  CO     6  CO
AS TR     7  PI     RE CO
8  DI     VA DI     10 DI
5  PI     AS PI     6  DI
8  CO     AS CO     DA CO
4  PI     6  TR     9  TR
3  DI     3  TR     RE TR
3  CO     5  TR     VA PI
DA PI     2  DI
```

PROGRAMA ZX SPECTRUM

```
  10 LET R$="234567891VDRA"
  15 LET R$=R$+R$+R$+R$
  20 LET U=9: PAPER 7: CLS
  30 LET B$="TRDICOPI0AAES"
  35 LET V=LEN B$
  40 FOR J=1 TO 4
  60 FOR I=1 TO V
  65 LET F=I-13*INT ((I-1)/13)
  66 LET X=INT (LEN R$*RND+1)
  75 LET S=2*INT ((X-1)/V)+1
  80 IF R$(X)="0" THEN GO TO 66
  81 LET T=X-V*INT (X/
  82 LET T=T+V*NOT T
  88 PRINT AT F,U*J-U;R$(X)+(B$(
T) AND T>=U);AT F,U*J-6; PAPER (
(S-1)/2)+3;B$(S TO S+1
 100 LET R$(X)="0"
 110 NEXT I
 120 BEEP 1,12: PAUSE 4E4
 140 NEXT J
```

```
2  TR     4  CO     10 DI     4  TR
6  TR     2  PI     AS PI     10 PI
VA DI     2  CO     6  DI     6  PI
4  PI     7  CO     3  DI     5  DI
2  DI     RE PI     RE CO     5  PI
VA TR     VA CO     8  DI     9  CO
DA TR     3  TR     8  CO     7  TR
9  TR     7  DI     9  DI     DA DI
DA CO     10 TR     9  PI     8  TR
5  TR     AS CO     AS DI     RE DI
10 CO     5  CO     AS TR     3  CO
VA PI     8  PI     7  PI     RE TR
6  CO     3  PI     DA PI     4  DI
```

Reloj 1

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: difícil.
Categoría del juego: dibujo.
Dificultad del juego: ninguna.

EL JUEGO: Transformar la pantalla del televisor en un reloj numérico con una visualización de gran tamaño gracias a este programa, que os permitirá animar una vitrina, crear un reloj de fantasía o servir para otros usos.

EL PROGRAMA: La línea 2Ø permitirá introducir la hora, bajo la forma de un entero de 4 cifras (por ejemplo: 1024 significa 10 horas 24 minutos).

La línea 5Ø realiza la conversión de cada cifra. Las líneas 6Ø a 13Ø efectúan el dibujo, retrasando la impresión poco a poco (con la ayuda de la variable K, incrementada en la línea 15Ø).

La línea 175 asegura el paso de las 24 horas 59 minutos a las 00 horas 00 minutos.

La línea 178 permite añadir 1 hora cada 60 minutos.

La línea 19Ø asegura la visualización (puede regularse la duración de la pausa para ajustar el reloj).

PARA JUGAR: Desde la puesta en marcha con RUN, el programa se sitúa en espera de la hora; se pulsa a la siguiente señal sonora, y se espera la señal para pulsar NEWLINE. Entonces se visualiza la hora bastante rápidamente, y el reloj se pone en marcha.

Para pararlo, se pulsará la tecla BREAK.

La hora se introducirá bajo la forma de un entero que tenga, necesariamente, 4 cifras; para las 5 horas y 6 minutos, se tendrá cuidado de pulsar 0506, seguido de NEWLINE.

POSIBLES EXTENSIONES: Será interesante modificar el programa, para confeccionar un despertador con alarma; también podrá mejorarse el grafismo, introducir color, utilizar la posibilidad del SPECTRUM de captar el tiempo, con el fin de tener una mayor precisión, etc.

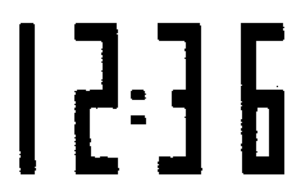
RELOJ 1

PROGRAMA ZX 81

```
  10 INPUT A$
  25 LET F=VAL "6"
  26 LET S=SGN PI
  27 LET G=F-S
  30 CLS
  40 LET K=S
  50 LET A=VAL A$(K)
  60 PRINT AT S,F*K-G;"███" AND
A<>S AND A<>G-S
  70 FOR J=S TO G
  80 PRINT AT J,F*K-G;"█" AND ((
A=NOT PI) OR (A>INT PI AND A<>F+
S));TAB F*K-INT PI;"█" AND A<>G
AND A<>F
  85 PRINT AT G+J-S,F*K-G;"█" AN
D (A=NOT PI OR A=S+S OR A=F OR A
=F+S+S);TAB F*K-INT PI;"█" AND A
<>S+S
  90 NEXT J
 100 PRINT AT G,F*K-G+S;"██" AND
 A>S;TAB F+G;"▀"
 130 PRINT AT F+INT PI,F*K-G;"██
█" AND A<>S AND A<>G-S AND A<>F+
S
 150 LET K=K+S
 170 IF K<G THEN GO TO CODE "M"
 172 LET A=S+VAL A$
 175 IF A=VAL "2360" THEN LET A=
NOT PI
 176 LET A$="000"+STR$ A
 177 LET A$=A$(LEN A$-INT PI TO
)
 178 IF A$(INT PI TO )="60" THEN
 LET A$=STR$ (VAL A$( TO S+S)+S)
+"00"
 190 PAUSE VAL "3000"
 210 GO TO CODE "2"
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "HORA ( FORMATO XXXX
)?": INPUT A$
  25 LET F=6
  26 LET S=1
  27 LET G=F-S
  30 BORDER 2: PAPER 6: CLS
  40 LET K=S
  50 LET A=VAL A$(K)
  60 PRINT INK 1;AT S,F*K-G;"███
" AND A<>S AND A<>G-S
  70 FOR J=S TO G
  80 PRINT INK 1;AT J,F*K-G;"█"
AND ((A=0) OR (A>3 AND A<>F+S));
AT J,F*K-3;"█" AND A<>G AND A<>F
  85 PRINT INK 1;AT G+J-S,F*K-G;
"█" AND (A=0 OR A=S+S OR A=F OR
A=F+S+S);AT G+J-S,F*K-3;"█" AND
A<>S+S
  90 NEXT J
 100 PRINT INK 1;AT G,F*K-G+S;"█
█" AND A>S
 130 PRINT INK 1;AT F+3,F*K-G;"█
██" AND A<>S AND A<>G-S AND A<>F
+S
 150 LET K=K+S
 170 IF K<G THEN GO TO 50
 172 LET A=S+VAL A$
 175 IF A=2360 THEN LET A=0
 176 LET A$="000"+STR$ A
 177 LET A$=A$(LEN A$-3 TO )
 178 IF A$(3 TO )="60" THEN LET
A$=STR$ (VAL A$( TO S+S)+S)+"00"
 179 LET H=S
 180 FOR J=S TO 115
 185 FOR I=S TO 16
 190 IF H THEN PRINT INK 1;AT G,
F+G;"▀";AT G+S,F+G;"▄"
 192 IF NOT H THEN PRINT AT G,F+
G;" ";AT G+S,F+G;" "
 195 NEXT I
 200 LET H=NOT H
 205 NEXT J
 210 BEEP .7,10: GO TO 30
```



El lobo y los corderos

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: difícil.

Categoría del juego: de reflexión (para dos jugadores).

Dificultad del juego: mediana.

EL JUEGO: Es un juego bastante conocido y que puede practicarse indistintamente en un tablero 8 × 8 o en uno de 10 × 10. Aquí se presenta la versión de un tablero de 8 casillas; uno de los jugadores escoge el lobo, que se desplaza una casilla en cada jugada, siguiendo las diagonales, quedándose en las casillas negras y que puede avanzar o retroceder.

Los corderos se desplazan igualmente por las casillas negras, en diagonal, pero no pueden retroceder. Están prohibidas las casillas que ya están ocupadas (no hay captura) y no existe el salto. Gana el lobo si consigue llegar a la línea de abajo antes de que todos los corderos ocupen la de arriba; ganan los corderos si sucede esto último o si consiguen bloquear al lobo.

EL PROGRAMA: Una variable A$, en la línea 2, contiene las posiciones de salida.

El bucle I (1Ø-4Ø) dibuja el tablero y las líneas 45 y 47 sitúan al lobo y a los corderos.

La línea 5Ø controla el cambio de jugador.

La línea 6Ø visualiza L cuando el turno es el del jugador que maneja al lobo, y C si se trata de su adversario.

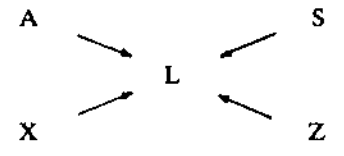
La línea 12Ø espera el movimiento del jugador.

La línea 13Ø escogerá el programa que analice la respuesta, en función del jugador de que se trate.

PARA JUGAR: RUN provoca la parada, en espera de una introducción: Ø y NEWLINE, si se quiere que empiecen los corderos, y 1 para el lobo. Después de esto, se visualiza el tablero y la frase TURNO C para los corderos y TURNO L para la jugada del lobo.

Los corderos responden con un número, seguido de D o de I (para derecha o izquierda). Así, 2D provoca el desplazamiento del cordero número 2, que sube en diagonal una casilla hacia la derecha.

Se visualiza el desplazamiento efectuado, y después se obtiene una pregunta idéntica para el lobo, que contestará tecleando A, S, X o Z, siguiendo la dirección escogida:



Deberá tenerse cuidado en no salir del tablero y no caer sobre el adversario, ya que no se ha previsto ningún control.

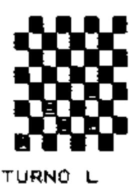
POSIBLES EXTENSIONES: **Podrán preverse unos puntos de referencia de las casillas, un grafismo más cuidado, controles para no pasarse de los límites del tablero o para no ir a parar a una casilla ya ocupada.**

También podría considerarse jugar con un tablero de 10 × 10 casillas, lo que aumentaría ligeramente la dificultad del juego de los corderos. De todas maneras, la ampliación más interesante que puede realizarse es, sin duda, la confección de un programa que permita al ZX comportarse como honroso adversario, ya sea como lobo o como corderos (y, en este caso, será posible definir una estrategia para que gane el ZX).

EL LOBO Y LOS CORDEROS

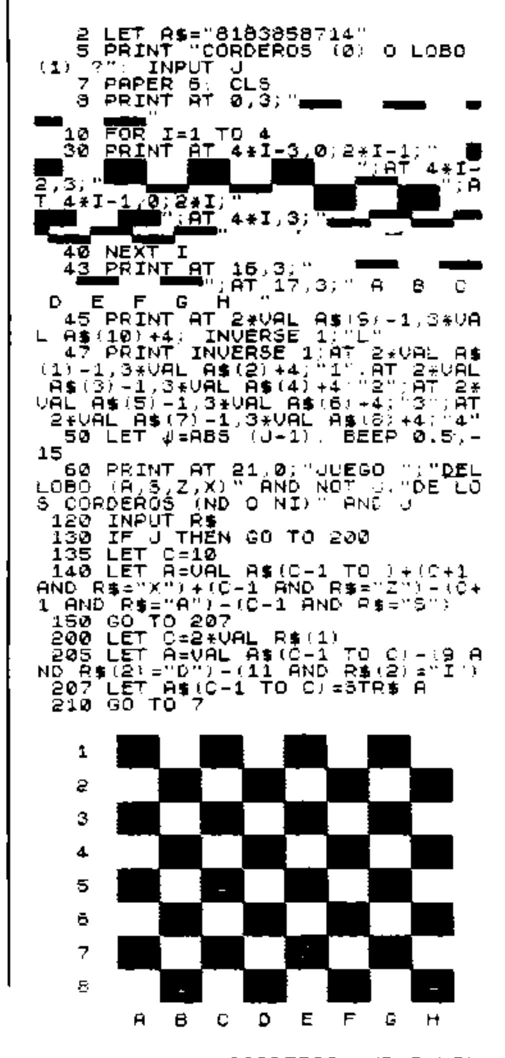
PROGRAMA ZX 81

```
  2 LET A$="8183858714"
  5 INPUT J
  7 CLS
 10 FOR I=SGN PI TO VAL "4"
 30 PRINT AT I+I,SGN PI;"█ █ █ █ ";AT I+I-SGN PI,SGN PI;" █ █ █ █"
 40 NEXT I
 45 PRINT AT VAL A$(9),VAL A$(10);"█"
 47 PRINT AT VAL A$(SGN PI),VAL A$(2);"█";AT VAL A$(INT PI),VAL A$(4);"█";AT VAL A$(5),VAL A$(6);"█";AT VAL A$(7),VAL A$(8);"█"
 50 LET J=ABS (J-SGN PI)
 60 PRINT AT PI*PI,NOT PI;"TURNO ";"L" AND NOT J;"C" AND J
120 INPUT R$
130 IF J THEN GO TO CODE "COS "
135 LET C=VAL "10"
140 LET A=VAL A$(C-SGN PI TO )+(C+SGN PI AND R$="X")+(C-SGN PI AND R$="Z")-(C+SGN PI AND R$="A")-(C-SGN PI AND R$="S")
150 GO TO 207
200 LET C=VAL "2"*VAL R$(SGN PI)
205 LET A=VAL A$(C-SGN PI TO C)-(VAL "9" AND R$(2)="D")-(VAL "11" AND R$(2)="I")
207 LET A$(C-SGN PI TO C)=STR$ A
210 GO TO VAL "7"
```



TURNO L

PROGRAMA ZX SPECTRUM



```
  2 LET A$="8183858714"
  5 PRINT "CORDEROS (0) O LOBO (1) ?": INPUT J
  7 PAPER 6: CLS
  8 PRINT AT 0,3;"▄▄▄   ▄▄▄   ▄▄▄   ▄▄▄"
 10 FOR I=1 TO 4
 30 PRINT AT 4*I-3,0;2*I-1;"  █  █  █  █ ";AT 4*I-2,3;"█  █  █  █  ";AT 4*I-1,0;2*I;"  █  █  █  █";AT 4*I,3;"▀▀▀   ▀▀▀   ▀▀▀   ▀▀▀"
 40 NEXT I
 43 PRINT AT 16,3;"   ▀▀▀   ▀▀▀   ▀▀▀   ▀▀▀";AT 17,3;" A  B  C  D  E  F  G  H "
 45 PRINT AT 2*VAL A$(9)-1,3*VAL A$(10)+4; INVERSE 1;"L"
 47 PRINT INVERSE 1;AT 2*VAL A$(1)-1,3*VAL A$(2)+4;"1".AT 2*VAL A$(3)-1,3*VAL A$(4)+4;"2";AT 2*VAL A$(5)-1,3*VAL A$(6)+4;"3";AT 2*VAL A$(7)-1,3*VAL A$(8)+4;"4"
 50 LET J=ABS (J-1): BEEP 0.5,-15
 60 PRINT AT 21,0;"JUEGO ";"DEL LOBO (A,S,Z,X)" AND NOT J;"DE LOS CORDEROS (ND O NI)" AND J
120 INPUT R$
130 IF J THEN GO TO 200
135 LET C=10
140 LET A=VAL A$(C-1 TO )+(C+1 AND R$="X")+(C-1 AND R$="Z")-(C+1 AND R$="A")-(C-1 AND R$="S")
150 GO TO 207
200 LET C=2*VAL R$(1)
205 LET A=VAL A$(C-1 TO C)-(9 AND R$(2)="D")-(11 AND R$(2)="I")
207 LET A$(C-1 TO C)=STR$ A
210 GO TO 7
```


JUEGO DE LOS CORDEROS (ND O NI)

Belote

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: mediana.
Categoría del juego: juego de cartas.
Dificultad del juego: fácil.

EL JUEGO: El juego que se propone se trata, más bien, de una simulación que se realiza con 32 cartas, con una distribución de 4 paquetes de 5 cartas. Podrá utilizarse para el estudio de problemas en los que intervengan distribuciones de 20 cartas sobre 32.

EL PROGRAMA: En la línea 1Ø se confecciona un tablero que pueda contener las 20 cartas.

El bucle I (2Ø-7Ø) asegura la tirada aleatoria de 20 números, tomados entre 32; el bucle J (4Ø-8Ø) verifica que no se realice dos veces la misma tirada.

El segundo bucle I (9Ø-18Ø) asegura, a continuación, la visualización; el nombre de las cartas se visualiza claramente, y el color se descifra y se visualiza en claro gracias a las líneas 117, 118, 12Ø y 123.

PARA JUGAR: Basta con pulsar RUN para obtener inmediatamente la visualización de 4 paquetes de 5 cartas cada uno (una distribución).

Pulsando de nuevo RUN, se obtiene una nueva distribución.

POSIBLES EXTENSIONES: Fácilmente podrá ponerse a punto una visualización más gráfica (sobre todo en el Spectrum); igualmente podrá utilizarse este programa como núcleo para la puesta a punto de programas de juegos de cartas por ordenador (el cual, si es necesario, podrá utilizarse como pareja: ver como ejemplo el PÓQUER en esta misma obra).

BELOTE

PROGRAMA ZX 81

```
  10 DIM A(CODE "*")
  15 LET K=SGN PI
  20 FOR I=K TO CODE "="
  30 LET A(I)=K+INT (CODE "4"*RN
D)
  40 FOR J=K TO I-K
  50 IF A(I)=A(J) THEN GO TO COD
E "2"
  60 NEXT J
  70 NEXT I
  90 FOR I=SGN PI TO CODE "="
 110 LET S$="DI"
 113 LET S=SGN PI+A(I)-VAL "8"*I
NT (A(I)/VAL "8")
 114 LET R$=STR$ (S+VAL "6")+"
"
 115 LET R$=R$( TO VAL "2")
 117 IF S=VAL "5" THEN LET R$="V
A"
 118 IF S=VAL "6" THEN LET R$="D
A"
 120 IF S=VAL "7" THEN LET R$="R
E"
 123 IF S=VAL "8" THEN LET R$="A
S"
 125 IF A(I)<=CODE "/" THEN LET
S$="TR"
 127 IF A(I)<=CODE "(" THEN LET
S$="PI"
 130 IF A(I)<=VAL "8" THEN LET S
$="CO"
 150 PRINT AT K,VAL "8"*INT ((I-
SGN PI)/VAL "5");R$;" ";S$
 160 LET K=K+SGN PI
 170 IF K=VAL "6" THEN LET K=SGN
 PI
 180 NEXT I
```

```
 RE PI    7  TR    8  DI    DA TR
 10 PI    AS CO    RE CO    AS TR
 7  PI    DA PI    7  DI    AS DI
 DA DI    10 CO    AS PI    VA CO
 RE TR    8  PI    9  PI    RE DI
```

PROGRAMA ZX SPECTRUM

```
  10 DIM A(20)
  15 LET K=1
  20 FOR I=K TO 20
  30 LET A(I)=K+INT (32*RND)
  40 FOR J=K TO I-K
  50 IF A(I)=A(J) THEN GO TO 30
  60 NEXT J
  70 NEXT I
  90 FOR I=1 TO 20
 110 LET S$="DI"
 113 LET S=1+A(I)-8*INT (A(I)/8)
 114 LET R$=STR$ (S+6)+"  "
 115 LET R$=R$( TO 2)
 117 IF S=5 THEN LET R$="VA"
 116 IF S=6 THEN LET R$="DA"
 120 IF S=7 THEN LET R$="RE"
 123 IF S=8 THEN LET R$="AS"
 125 IF A(I)<=24 THEN LET S$="TR
"
 127 IF A(I)<=16 THEN LET S$="PI
"
 130 IF A(I)<=8 THEN LET S$="CO"
 150 PRINT AT K+K,8*INT ((I-1)/5
);R$;" "; PAPER INT ((A(I)-1)/8)
+3; BRIGHT 1;S$: BEEP 0.1,15
 160 LET K=K+1
 170 IF K=6 THEN LET K=1
 180 NEXT I
```

```
7  CO     RE CO     9  PI     10 TR

VA CO     AS DI     8  CO     7  PI

9  CO     8  PI     DA DI     AS TR

VA TR     8  TR     7  DI     10 DI

7  TR     AS PI     10 CO     DA TR
```

Un cálculo fastidioso

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: de reflexión (para dos jugadores).
Dificultad del juego: difícil.

EL JUEGO: Se practica con dos jugadores; uno realiza la función de codificador y el otro la de decodificador.

El codificador parte de un número (propuesto por el ZX) y debe transformarlo con la ayuda de 5 operaciones que él mismo escoge, así como los operadores.

El resultado que ha obtenido y las operaciones utilizadas se visualizan en la pantalla; el decodificador deberá entonces intentar encontrar el número inicial, efectuando las operaciones en sentido contrario.

EL PROGRAMA: La línea 5 permite obtener el entero inicial, comprendido entre 5 y 20.

Un bucle I (5Ø-1ØØ) permite introducir sucesivamente las 5 operaciones.

Se utilizan 4 subprogramas para los cálculos (en las líneas 1ØØØ a 4ØØØ).

Después de cada cálculo, se visualiza el resultado (solamente el codificador deberá mirar la pantalla durante esta operación), y el resultado definitivo quedará visualizado al final de los cálculos (línea 95), para el decodificador.

La línea 14Ø visualiza las 5 operaciones utilizadas, así como los operadores correspondientes.

PARA JUGAR: Se pulsa RUN para poner en marcha el programa.

Se visualiza el número inicial, después la pregunta INTRODUCIR 5 OPERACIONES; deberán introducirse una después de la otra, bajo la forma, por ejemplo: + 4, seguido de NEWLINE.

El resultado se visualiza después de cada operación introducida. Al final, se visualizan en una misma línea todas las operaciones introducidas y se envía al decodificador el mensaje ÁNIMO (no existe control para la decodificación).

POSIBLES EXTENSIONES: Desde luego, podrá mejorarse la parte de decodificación del programa introduciendo controles en el resultado; también puede considerarse dejar participar más tiempo al ZX, confiándole el cometido de la codificación, etc.

UN CÁLCULO FASTIDIOSO

PROGRAMA ZX 81

```
   3 CLS
   5 LET X=INT (RND*16+5)
  10 PRINT "NUMERO INICIAL ";X
  20 LET L=SGN PI
  30 PRINT "INTRODUCIR 5 OPERACI
ONES"
  35 LET F=X
  40 PRINT "FORMATO +6 , /5 . ..
 "
  45 DIM R$(5,INT PI)
  50 FOR I=L TO 5
  60 INPUT R$(I)
  70 LET A=CODE R$(I,L)-20
  80 IF A<=NOT PI OR A>4 THEN GO
 TO 60
  90 GO SUB A*1000
  95 PRINT AT 4,L;"NUMERO OBTENI
DO ";F;"  "
 100 NEXT I
 130 PRINT "OPERACIONES"
 140 PRINT R$(4);TAB 5;R$(1);TAB
 10;R$(5);TAB 15;R$(3);TAB 20;R$
(2)
 150 PRINT "ANIMO"
 160 PAUSE 4E4
 170 RUN
1000 LET F=F*VAL R$(I,2 TO )
1010 RETURN
2000 LET F=F+VAL R$(I,2 TO )
2010 RETURN
3000 LET F=F-VAL R$(I,2 TO )
3010 RETURN
4000 LET F=F/VAL R$(I,2 TO )
4010 RETURN
```

```
NUMERO INICIAL 18
INTRODUCIR 5 OPERACIONES
FORMATO +6 , /5 , ...

 NUMERO OBTENIDO 0
OPERACIONES
/4   -5   -10  +14  *2
ANIMO
```

PROGRAMA ZX SPECTRUM

```
   3 CLS
   5 LET X=INT (RND*16+5)
  10 PRINT INVERSE 1;"NUMERO INI
CIAL "; INVERSE 0;" "; FLASH 1;X
  20 LET L=1
  30 PRINT "INTRODUCIR 5 OPERACI
ONES"
  35 LET F=X
  40 PRINT "FORMATO +6 , /5 . ..
."
  45 DIM R$(5,3)
  50 FOR I=L TO 5
  60 INPUT R$(I)
  70 LET A=CODE R$(I,L)-41
  80 IF A<=0 OR A=3 OR A=5 OR A>
6 THEN GO TO 60
  85 IF A>3 THEN LET A=(A/2)+1
  90 GO SUB A*1000
  95 PRINT INVERSE 1;AT 4,L;"NUM
ERO OBTENIDO "; INVERSE 0;"  ";
FLASH 1; PAPER 5;F; FLASH 0; PAP
ER 7;"  "
 100 NEXT I
 130 PRINT "OPERACIONES"
 140 PRINT PAPER 4;R$(4);TAB 5;R
$(1);TAB 10;R$(5);TAB 15;R$(3);T
AB 20;R$(2)
 150 PRINT "ANIMO"
 160 PAUSE 4E4
 170 RUN
1000 LET F=F*VAL R$(I,2 TO )
1010 RETURN
2000 LET F=F+VAL R$(I,2 TO )
2010 RETURN
3000 LET F=F-VAL R$(I,2 TO )
3010 RETURN
4000 LET F=F/VAL R$(I,2 TO )
4010 RETURN
```

```
NUMERO INICIAL 18
INTRODUCIR 5 OPERACIONES
FORMATO +6 , /5 , ...

 NUMERO OBTENIDO    35
OPERACIONES
/6   -17  +25  +7   *53
ANIMO
```

Póquer

REFERENCIAS:

Modalidad: FAST.

Dificultad de programación: difícil.

Categoría del juego: juego contra el ZX.

Dificultad del juego: fácil.

EL JUEGO:

Se trata de la simulación de una partida tradicional de PÓQUER, realizada con un juego de 32 cartas, en la que el jugador tendrá como adversario al ZX.

El jugador recibe 5 cartas, referenciadas por números del 1 al 5, y puede decidir el cambio de algunas de ellas, anunciando las que descarta.

Se visualiza el nuevo juego, y a continuación es el turno del ZX.

EL PROGRAMA:

Las líneas 1Ø y 2Ø definen las cartas (con 1 en lugar de 10 y A en lugar de AS).

Una variable N se dimensiona en 15, en la línea 3Ø, para recibir las 10 cartas del jugador y las 5 del ZX.

Un bucle I (4Ø-9Ø) efectúa la tirada, y un bucle J (6Ø-8Ø) realiza el control.

El subprograma 229-1Ø6Ø efectúa la visualización de las cartas.

Otro bucle I (15Ø-17Ø) realiza la segunda tirada del jugador; la línea 18Ø permite la segunda visualización, y el bucle I (229-1Ø5Ø), donde después se le introduce directamente, realizará la visualización del juego de la máquina.

PARA JUGAR:

La orden RUN pone en marcha la ejecución del programa, y se visualizan las 5 primeras cartas del jugador.

A la pregunta DESCARTE, el jugador puede contestar Ø seguido de NEWLINE, si quiere terminar la partida.

Para jugar, responderá simplemente NEWLINE si considera que está servido o introducirá los números de las cartas que rechaza, seguido de NEWLINE.

Por ejemplo, si teclea 235 y NEWLINE, su nuevo juego conservará las cartas 1 y 4.

Entonces se visualizará su juego, así como el del ZX; este último no tiene, en este programa, la posibilidad de volver a tomar cartas.

A continuación, bastará pulsar NEWLINE para empezar una nueva partida.

POSIBLES EXTENSIONES: Desde luego serán muy numerosas, desde una visualización gráfica en color de las cartas hasta una mejora en el juego del ZX, introducir reglas para apostar y hasta, ¿por qué no?, posibilidades de farol.

PÓQUER

PROGRAMA ZX 81

```
   1 CLS
   5 LET S=VAL "5"
  10 LET V$=" 7 8 910VADAREAS"
  20 LET C$="TRDICOPI"
  30 DIM N(VAL "15")
  40 FOR I=SGN PI TO VAL "15"
  50 LET N(I)=INT (RND*CODE "4")
  60 FOR J=SGN PI TO I-SGN PI
  70 IF N(I)=N(J) THEN GO TO COD
E "M"
  80 NEXT J
  90 NEXT I
 100 LET K=NOT PI
 105 PRINT AT S*PI,NOT PI;"CARTA
S RECHAZADAS ?"
 110 GO SUB CODE " FAST "
 140 IF R$="0" THEN STOP
 150 FOR I=SGN PI TO LEN R$
 160 LET N(VAL R$(I))=N(I+5-SGN
PI)
 170 NEXT I
 175 CLS
 180 GO SUB CODE " FAST "
 200 LET K=VAL "8"
 229 FOR I=SGN PI TO S-SGN PI
1010 LET J=VAL "2"*INT (N(I+K)/V
AL "8")+SGN PI
1015 LET F=VAL "2"*(N(I+K)-VAL "
8"*INT (N(I+K)/VAL "8"))+SGN PI
1020 LET R$=V$(F TO F+SGN PI)
1030 PRINT AT K,S*I-PI CHR$ (COD
E "■"+I);AT K+SGN PI,S*I-3+SGN P
I;R$;" ";C$(J TO J+SGN PI)
1050 NEXT I
1055 INPUT R$
1058 IF K THEN RUN
1060 RETURN
```

```
AS TR VA DI  8 TR  8 CO  8 PI

10 TR DA DI RE PI AS PI AS CO
```

PROGRAMA ZX SPECTRUM

```
   1 CLS
   5 LET S=6
  10 LET V$="7 8 9 10VADAREAS"
  20 LET C$="TRDICOPI"
  30 DIM N(15)
  40 FOR I=1 TO 15
  50 LET N(I)=INT (RND*32)
  60 FOR J=1 TO I-1
  70 IF N(I)=N(J) THEN GO TO 50
  80 NEXT J
  90 NEXT I
 100 LET K=0
 105 PRINT AT 21,0;"CARTAS RECHA
ZADAS ?"
 110 GO SUB 229
 140 IF R$="0" THEN STOP
 150 FOR I=1 TO LEN R$
 160 LET N(VAL R$(I))=N(I+5-1)
 170 NEXT I
 175 CLS
 180 GO SUB 229
 200 LET K=10
 229 FOR I=1 TO 5
1010 LET J=2*INT (N(I+K)/8)+1
1015 LET F=2*(N(I+K)-8*INT (N(I+
K)/8))+1
1020 LET R$=V$(F TO F+1)
1030 PRINT AT K,S*I-4; INVERSE 1
;I; INVERSE 0;AT K+2,S*I-5;R$;":
"; PAPER (J+1)/2+1; BRIGHT 1;C$(
J TO J+1): BEEP .1,0: PRINT AT K
+3,S*I-5;"▬▬▬▬▬"
1050 NEXT I
1055 INPUT R$
1058 IF K THEN RUN
1060 RETURN
```

```
VA:PI DA:PI AS:TR VA:CO VA:DI

9 :CO DA:DI VA:TR RE:PI RE:TR
```

Hanoï

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: difícil.
Categoría del juego: de reflexión.
Dificultad del juego: difícil.

EL JUEGO: Se presenta aquí una versión para un solo jugador. Se dispone de un cierto número de discos (ocho como máximo), y hay que transferirlos desde la torre número 1 a la torre número 3, respetando una sola regla: en ningún momento, un disco podrá estar recubierto por otro disco de un diámetro superior.

Este programa no contabiliza el número de jugadores; sin embargo, no se aceptarán las jugadas no válidas.

EL PROGRAMA: La variable D$, en la línea 4Ø, servirá para dibujar los discos.

El número de discos se introduce en la línea 6Ø, y se rechazará si hay más de 8 discos.

El buche I (7Ø-9Ø) permite confeccionar una variable A (I), que contiene el estado de la pila en forma numérica (en la línea 5 están previstas tres variables numéricas, para conservar el estado de las tres pilas).

La línea 11Ø dibuja los zócalos, y los bucles intercalados I (12Ø-16Ø) y J (125-15Ø) dibujan las tres pilas.

La línea 23Ø realiza el control de la jugada, y las líneas 24Ø y 25Ø modifican el estado de las pilas.

PARA JUGAR: Después de RUN, que pone en marcha el programa, se introduce el número de discos, seguido de NEWLINE.

Se obtiene el dibujo del estado inicial, y a la pregunta SAL., se responde dando el número de la torre de salida. No se aceptará este número (1, 2 o 3) si la torre está vacía. Después, se pulsará NEWLINE, y a la pregunta LLE., se contestará con el número de la torre de llegada. Si la jugada no es posible, será obligatorio volver a empezar en SAL.

Si la jugada es posible, se realizará el desplazamiento y se vuelve a empezar.

POSIBLES EXTENSIONES: Podrán dibujarse distintos colores, introducir un contador de jugadas y, sobre todo, definir un juego para el ordenador. También puede intentarse complicar el juego con dos varillas y dos jugadores, cada uno con discos de distinto color, por ejemplo.

HANOÏ

PROGRAMA ZX 81

```
   5 DIM A(INT PI)
   7 LET U=INT (PI*PI)
  10 LET A(SGN PI)=U
  20 LET A(SGN PI+SGN PI)=U
  30 LET A(INT PI)=U
  40 LET D$="[redacted]"
  50 PRINT "NUMERO"
  60 INPUT N
  65 IF N>=U-SGN PI THEN RUN
  70 FOR I=NOT PI TO N-SGN PI
  80 LET A(SGN PI)=VAL (STR$ A(S
GN PI)+STR$ (N-I))
  90 NEXT I
 100 CLS
 105 LET C$="[redacted]"
 110 PRINT AT PI*PI,NOT PI;C$;"1
";C$;"2 ";C$;"3"
 120 FOR I=NOT PI TO N-SGN PI
 125 FOR J=SGN PI TO INT PI
 127 LET B$=STR$ A(J)
 130 IF LEN B$>I+SGN PI THEN PRI
NT AT U-I,(U+SGN PI)*J-U;D$( TO
VAL B$(I+SGN PI+SGN PI))
 150 NEXT J
 160 NEXT I
 170 PRINT AT CODE "$",NOT PI;"S
AL"
 180 INPUT D
 185 IF A(D)=U THEN GOTO CODE "[redacted]
"

 190 PRINT "LLE"
 200 INPUT A
 210 LET B$=STR$ A(D)
 220 LET C$=STR$ A(A)
 230 IF B$(LEN B$)>C$(LEN C$) TH
EN GOTO CODE "[redacted]"
 240 LET A(A)=VAL (C$+B$(LEN B$)
)
 250 LET A(D)=VAL B$( TO LEN (B$
)-SGN PI)
 260 GOTO 100
```




```
SAL
LLE
```

PROGRAMA ZX SPECTRUM

```
   5 DIM A(3)
   7 LET U=9
  10 LET A(1)=U
  20 LET A(2)=U
  30 LET A(3)=U
  40 LET D$="[redacted]"
  50 PRINT "NUMERO DE TORRES (<8
) :";
  60 INPUT N: PRINT N
  65 IF N>=8 THEN RUN
  70 FOR I=0 TO N-1
  80 LET A(1)=VAL (STR$ A(1)+STR
$ (N-I))
  90 NEXT I
 100 CLS
 105 LET C$="[redacted]"
 110 PRINT AT 10,0; INK 1;C$;TAB
 10;C$;TAB 20;C$
 115 PRINT AT 12,4; INK 1;"1";TA
B 14;"2";TAB 24;"3"
 120 FOR I=0 TO N-1
 125 FOR J=1 TO 3
 127 LET B$=STR$ A(J)
 130 IF LEN B$>I+1 THEN PRINT AT
 U-I,(U+1)*J-U; INK 2;D$( TO VAL
 B$(I+2))
 150 NEXT J
 160 NEXT I
 170 PRINT AT 14,0;"TORRE DE SAL
IDA :  ";
 180 INPUT D: PRINT AT 14,17;D
 185 IF A(D)=U THEN BEEP 1,0: GO
 TO 170
 190 PRINT AT 15,0;"TORRE DE LLE
GADA :  ";
 200 INPUT A: PRINT AT 15,17;A
 210 LET B$=STR$ A(D)
 220 LET C$=STR$ A(A)
 230 IF B$(LEN B$)>C$(LEN C$) TH
EN BEEP 1,0: GO TO 170
 240 LET A(A)=VAL (C$+B$(LEN B$)
)
 250 LET A(D)=VAL B$( TO LEN (B$
)-1)
 260 BEEP .2,10: GO TO 100
```



Música

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: juego educativo.

Dificultad del juego: mediana.

EL JUEGO: Se presenta como un ejercicio de aprendizaje del solfeo: el aparato anuncia una nota (en clave de sol), seguida de la mención GRAVE o AGUDA, según su emplazamiento en el pentagrama.

Se responderá indicando sobre qué línea se visualizará la nota.

Se dibuja entonces la nota en el pentagrama, seguida del mensaje BRAVO o HA PERDIDO, según se haya acertado o no.

EL PROGRAMA: El bucle I (1Ø-3Ø) dibuja el pentagrama.

Un segundo bucle I (4Ø-6Ø) visualiza los números de las líneas.

La línea 7Ø define el nombre de las notas, y la línea 8Ø realiza la extracción aleatoria de una de ellas.

El nombre de la nota se calcula en la línea 85 y se visualiza con la línea 9Ø, seguida de la mención GRAVE o AGUDA, según el caso.

La respuesta se introducirá en la línea 11Ø.

La línea 115 se ocupa del desciframiento de la respuesta.

La línea 12Ø visualiza la nota.

La línea 13Ø comprueba si se ha ganado o se ha perdido y visualiza el mensaje correspondiente.

Las líneas 14Ø y 15Ø permiten hacer un nuevo intento.

PARA JUGAR: Se pone en marcha el programa con RUN.

Se obtiene entonces la visualización del pentagrama, así como el nombre de la nota que tiene que situarse y la pregunta LÍNEA.

Se responde dando un número de línea (de 1 a E), seguido de NEWLINE.

Se dibuja entonces la nota, y el mensaje BRAVO o HA PERDIDO aparece en este caso en la pantalla.

Puede pulsarse cualquier tecla para continuar, y se terminará pulsando BREAK.

POSIBLES EXTENSIONES: Puede ampliarse a otra clave, además de la de sol, así como introducirse color, sonido, etc.

MÚSICA

PROGRAMA ZX 81

```
   5 CLS
  10 FOR I=SGN PI TO 5
  20 PRINT AT 4+I+I,INT PI;"█████
███████████"
  30 NEXT I
  40 FOR I=SGN PI TO 14
  50 PRINT AT I+2,NOT PI;CHR$ (2
8+I)
  60 NEXT I
  70 LET N$="SI LA SOLFA MI RE D
O "
  80 LET X=INT (RND*14+SGN PI)
  85 LET T=X-SGN PI-7*INT ((X-SG
N PI)/7)
  90 PRINT AT CODE ">",NOT PI;"N
OTA ";N$(INT PI*T+SGN PI TO INT
PI*T+INT PI);" AGUDA" AND X<=7;"
 GRAVE" AND X>7
 100 PRINT "LINEA"
 110 INPUT L$
 115 LET L=CODE L$-28
 120 PRINT AT X,PI*PI;"▌";AT X+S
GN PI,PI*PI;"▌";AT X+2,PI*PI;"█"
 130 PRINT AT CODE "=",NOT PI;"B
RAVO" AND L=X;"HA PERDIDO" AND L
<>X
 140 PAUSE 4E4
 150 RUN
```



PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 FOR I=1 TO 5
  20 PRINT INK 1;AT 4+I+I,3;"████
████████████"
  30 NEXT I
  40 FOR I=1 TO 14
  50 PRINT AT I+2,0;I AND I<=9;A
T I+2,0;CHR$ (55+I) AND I>9
  60 NEXT I
  70 LET N$="SI LA SOLFA MI RE D
O "
  80 LET X=INT (RND*14+1)
  85 LET T=X-1-7*INT ((X-1)/7)
  90 PRINT AT 18,0;"NOTA ";N$(3*
T+1 TO 3*T+3);" AGUDA" AND X<=7;
" GRAVE" AND X>7
 100 PRINT "LINEA :";
 110 INPUT L$: PRINT L$
 115 LET L=(CODE L$-48)*(L$<":")
+(CODE L$-55)*(L$>"@")
 120 PRINT INVERSE 1; INK 2;;AT
X,10;"▌";AT X+1,10;"▌";AT X+2,10
;"O": BEEP 1,20-X
 130 PRINT AT 20,0;"BRAVO" AND L
=X;"HA PERDIDO" AND L<>X
 140 PAUSE 4E4
 150 RUN
```



Ahorcado al revés

REFERENCIAS:

Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: juego sobre palabras.

Dificultad del juego: fácil.

EL JUEGO:

El ahorcado es un juego de letras muy conocido, que ya ha sido presentado en esta obra; en él se trata de adivinar una palabra escondida por el ordenador, que a cada letra no descubierta va acercando el jugador a la horca. Aquí, es el ZX el que tratará de adivinar la palabra, pero juega un poco al azar y no posee la intuición del jugador; así que, por regla general, tendrá que efectuar muchos intentos para descubrir la solución. Para aumentar sus posibilidades, se ha buscado una forma que le permita utilizar mucho más las vocales, después las consonantes más corrientes y, por último, las consonantes más raras.

EL PROGRAMA:

La línea 1Ø permite grabar en una variable alfanumérica B$ las letras del alfabeto en el orden: vocales, consonantes corrientes y otras consonantes. La palabra que hay que adivinar se introduce en la línea 5Ø (de hecho, basta con introducir su longitud).

El bucle I (6Ø-9Ø) imprime las rayas en el emplazamiento de las letras que hay que encontrar.

La pausa, en la línea 1Ø5, sirve únicamente para simular el tiempo de búsqueda de una letra.

La fórmula, situada en la línea 11Ø, permite modificar ligeramente la probabilidad de la extracción en favor de las vocales y de las consonantes corrientes.

La línea 13Ø provoca una nueva extracción en el caso de que la letra ya hubiera sido probada.

La línea 16Ø visualiza la letra escogida, y el bucle I (17Ø-2ØØ) vuelve a imprimir las rayas y las letras encontradas.

La línea 2Ø5 incrementa el contador de intentos.

La línea 21Ø visualiza, al final de la partida, el número de intentos realizados.

PARA JUGAR:

Poner en marcha el programa con RUN.

Introducir la palabra que hay que descubrir, seguido de NEWLINE.

Entonces se visualizan las rayas (una por letra), y el ZX propone una primera letra, mira si le conviene y continúa así automáticamente hasta que la haya encontrado.

Se puede terminar el programa en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Sin duda, será interesante para los niños prever una versión no automática, en la que el jugador respondería SÍ o NO a cada letra propuesta por el ZX; también podrá añadirse una visualización gráfica, por ejemplo, una horca o un coche avanzando hacia un precipicio, etc.

Con un minidisquete, el ZX podrá guardar las palabras utilizadas más corrientemente por los jugadores.

AHORCADO AL REVÉS

PROGRAMA ZX 81

```
  10 LET D$="YAEIOUBCDFGJLMNPRST
VWHKQXZ"
  20 LET C=NOT PI
  30 LET T$=""
  40 PRINT "SU PALABRA"
  50 INPUT R$
  60 FOR I=SGN PI TO LEN R$
  70 PRINT AT 5,I;"-"
  80 LET T$=T$+"-"
  90 NEXT I
 100 PRINT AT PI*PI,NOT PI;"BUSC
O...";" INTENTO ";C
 105 PAUSE CODE "█"
 110 LET X=6*(RND<.7)+13*(RND<.5
)+7*(RND<.4)
 120 LET X=INT (X*RND+SGN PI)
 130 IF D$(X)="-" THEN GO TO 110
 140 LET E$=D$(X)
 150 LET D$(X)="-"
 160 PRINT AT CODE "$" NOT PI;"T
OMO :";E$
 170 FOR I=SGN PI TO LEN R$
 180 IF E$=R$(I) THEN LET T$(I)=
E$
 190 PRINT AT 5,I;T$(I)
 200 NEXT I
 205 LET C=C+SGN PI
 210 IF T$=R$ THEN PRINT AT CODE
 "=",NOT PI;"ENCONTRADO EN ";C-S
GN PI;" VECES";W
 230 GO TO 100
```

```
SU PALABRA

 ---

BUSCO... INTENTO 0

TOMO :E
```

PROGRAMA ZX SPECTRUM

```
  10 LET D$="YAEIOUBCDFGJLMNPRST
VWHKQXZ"
  20 LET C=0
  30 LET T$=""
  40 PRINT "SU PALABRA"
  50 INPUT R$
  60 FOR I=1 TO LEN R$
  70 PRINT AT 5,I;"-"
  80 LET T$=T$+"-"
  90 NEXT I
 100 PRINT INVERSE 1; INK 1;AT 1
0,0;"BUSCO...";" INTENTO ";C
 105 PAUSE 150
 110 LET X=6*(RND<.7)+13*(RND<.5
)+7*(RND<.4)
 120 LET X=INT (X*RND+1)
 130 IF D$(X)="-" THEN GO TO 110
 140 LET E$=D$(X)
 150 LET D$(X)="-"
 160 PRINT AT 15,0; INVERSE 1; I
NK 2;"TOMO :"; FLASH 1;E$: BEEP
1,10
 170 FOR I=1 TO LEN R$
 180 IF E$=R$(I) THEN LET T$(I)=
E$
 190 PRINT AT 5,I;T$(I
 200 NEXT I
 205 LET C=C+1
 210 IF T$=R$ THEN PRINT INVERSE
 1; BRIGHT 1; INK 4;AT 20,0;"ENC
ONTRADO EN ";C-1;" VECES": BEEP
1,12: BEEP 1,18: BEEP 1,6: STOP
 230 GO TO 100
```

```
SU PALABRA

 PROGRAMA
```



Mateo

REFERENCIAS:

Modalidad: SLOW o FAST.

Dificultad de programación: fácil.

Categoría del juego: dibujo aleatorio.

Dificultad del juego: ninguna.

EL JUEGO:

Se utilizan la definición de una curva en coordenadas polares y la función PLOT para obtener en la pantalla dibujos aleatorios (del tipo flor), más o menos elegantes (el PLOT del ZX Spectrum es, por supuesto, mucho más preciso que el del ZX 81).

Los dos parámetros principales, A y B, podrán, si es preciso, introducirse a mano (instrucciones INPUT) para hacer un estudio más sistemático de las distintas curvas posibles.

EL PROGRAMA:

La línea 1Ø permite definir el radio máximo, y la línea 2Ø corrige este parámetro a fin de utilizar la superficie de la pantalla lo mejor posible.

Evidentemente, podrán modificarse estos dos parámetros, teniendo en cuenta que queden dentro de los límites de la pantalla.

Las líneas 3Ø y 4Ø definen dos variables aleatorias A y B, escogidas aquí entre 0 y 3; el trazado de las curvas dependerá de estas variables; podrá modificarse el intervalo de los valores o incluso introducirlos a mano escribiendo 3Ø INPUT A y 4Ø INPUT B.

El bucle I (1ØØ-145) realiza el trazado punto por punto; el radio R se calcula en la línea 11Ø, y las líneas 12Ø y 13Ø efectúan el paso en coordenadas cartesianas (imprescindibles para PLOT).

La línea 14Ø realiza la impresión de los puntos.

La línea 15Ø visualiza el mensaje ACABADO.

Las líneas 16Ø y 17Ø permiten la ejecución de otro dibujo, tan pronto como se pulse una tecla.

PARA JUGAR:

Se pone en marcha el programa pulsando RUN.

El dibujo se ejecuta automáticamente.

El mensaje ACABADO (abajo, a la izquierda) anuncia el final del trazado.

Para realizar otro dibujo pulsar cualquier tecla.

En la versión Spectrum, debe pulsarse Ø para borrar el dibujo anterior (si no, el nuevo dibujo se superpondrá al antiguo).

Se pulsará BREAK para salir del programa.

POSIBLES EXTENSIONES: Podrán utilizarse otras muchas funciones para realizar dibujos aún más extraordinarios (por ejemplo, dibujos que simulen una tercera dimensión).

MATEO

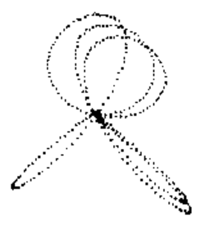
PROGRAMA ZX 81

```
   5 CLS
  10 LET H=3
  20 LET K=6
  30 LET A=3*RND
  40 LET B=3*RND
 100 FOR I=0 TO PI*PI STEP 3*PI/
180
 110 LET R=H*SIN (3*I)
 120 LET X=30+INT (K*R*COS (A*I)
)
 130 LET Y=24+INT (K*R*SIN (B*I)
)
 140 PLOT X,Y
 145 NEXT I
 150 PRINT AT 20,0;"ACABADO"
 160 PAUSE 4E4
 170 RUN
```

ACABADO

PROGRAMA ZX SPECTRUM

```
   5 BORDER 5: BRIGHT 1: CLS
  10 LET Z=7
  20 LET K=8
  30 LET T=3
  40 LET A=3*RND: LET B=3*RND
  50 INK INT (4*RND+1)
 100 FOR H=0 TO PI*2 STEP PI/180
 110 LET R=Z*SIN (H*T)
 120 LET X=120+INT (K*R*COS (A*H
))
 130 LET Y=90+INT (K*R*SIN (B*H)
)
 150 PLOT X,Y
 160 NEXT H
 165 PRINT AT 20,0;"ACABADO"
 170 BEEP .6,10: PAUSE 4E4
 175 PRINT AT 20,0;"       "
 180 IF INKEY$="0" THEN RUN
 190 GO TO 40
```



ACABADO

Máquina de palabras

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: fácil.
Categoría del juego: juego de azar (sobre palabras).
Dificultad del juego: ninguna.

EL JUEGO: Este programa tan corto permite al ZX demostrar su capacidad para manejar cadenas de caracteres: se parte simplemente de un prefijo y de un sufijo de origen griego, los cuales se ensamblarán al azar. Se caerá por supuesto, de vez en cuando, en palabras absurdas o hasta ligeramente equívocas.

En cada prueba se confecciona una palabra nueva, aunque también es posible obtener varias veces la misma palabra.

EL PROGRAMA: La línea 1Ø permite introducir la lista de prefijos en una variable A$ (naturalmente podrán incorporarse otros prefijos, que no tengan más de 5 letras, añadiendo blancos en caso necesario).

La línea 2Ø contendrá, de la misma forma, la lista de sufijos, que se sitúan en otra variable alfanumérica B$.

La línea 3Ø realiza la elección al azar de un prefijo.

La línea 4Ø efectúa la elección al azar de un sufijo.

La línea 5Ø visualiza la palabra obtenida, y la línea 7Ø permitirá volver a empezar la jugada.

PARA JUGAR: Poner en marcha la ejecución del programa con RUN.
Se visualiza inmediatamente una primera palabra.

Para obtener una nueva palabra, se pulsará cualquier tecla (menos BREAK); entonces se visualizará otra palabra, a continuación de la anterior.

Pulsar BREAK para salir del programa.

POSIBLES EXTENSIONES: Fácilmente podrá aumentarse la lista de palabras disponibles (en el ZX Spectrum, será preferible utilizar instrucciones DATA, que servirán para que no se produzcan problemas con la longitud de las palabras).

También podrá utilizarse un procedimiento análogo para que construya frases el ordenador, hacer que invente poemas, etc.

También podrá transformarse el programa inicial en un juego de sociedad que consistiría, por ejemplo, en hacer apuestas sobre la próxima palabra que tenga que aparecer. Y, eventualmente, confiar la gestión de las apuestas al ZX.

MÁQUINA DE PALABRAS

PROGRAMA ZX 81

```
  10 LET A$="ANDROCOSMOMICROHELI
O PEDOPORNOPALEOMACROHYDROSTENOP
ATHO OLEOMYTHOHOMEO OCTO"
  20 LET B$="CRATEPHILEPHOBETROP
EGYNE MANE NAUTEPHAGEFUGE PATHEC
ARDEDERMESCOPEVORE PHONE"
  30 LET I=5*INT ((LEN A$/5)*RND
)+SGN PI
  40 LET J=5*INT ((LEN B$/5)*RND
)+SGN PI
  50 PRINT A$(I TO I+4)+B$(J TO
J+4)
  60 PAUSE 4E4
  70 GO TO 30
```

```
STENOSCOPE
 OLEOPATHE
MICRODERME
HOMEOPHOBE
COSMOPHONE
HOMEOFUGE
ANDROVORE
PORNOPHONE
 OLEONAUTE
COSMOFUGE
 OLEOPHILE
HYDROCARDE
STENOFUGE
HELIOCRATE
PATHOSCOPE
PORNOVORE
MYTHOSCOPE
STENOPHAGE
PALEOGYNE
```

PROGRAMA ZX SPECTRUM

```
  10 LET A$="ANDROCOSMOMICROHELI
O PEDOPORNOPALEOMACROHYDROSTENOP
ATHO OLEOMYTHOHOMEO OCTO"
  20 LET B$="CRATEPHILEPHOBETROP
EGYNE MANE NAUTEPHAGEFUGE PATHEC
ARDEDERMESCOPEVORE PHONE"
  30 LET I=5*INT ((LEN A$/5)*RND
)+1
  40 LET J=5*INT ((LEN B$/5)*RND
)+1
  50 PRINT INVERSE 1; INK INT (5
*RND+1);A$(I TO I+4)+B$(J TO J+4
)
  60 BEEP .3,18: PAUSE 4E4
  70 GO TO 30
```



Componer

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: fácil.
Categoría del juego: composición musical.
Dificultad del juego: ninguna, *a priori.*

EL JUEGO: Se trata, en efecto, de un programa capaz de componer música de una manera muy elemental, de forma completamente aleatoria. El ordenador, en un primer momento, dibujará un pentagrama, después visualizará una sucesión de notas (que van del DO grave al SI agudo) y dará su nombre. Se visualiza a continuación una nueva serie de notas, y así sucesivamente.

EL PROGRAMA: El bucle I (1Ø-3Ø) dibuja el pentagrama.
La variable N$, en la línea 4Ø, contiene el nombre de las notas.
En la línea 6Ø, el segundo bucle I (5Ø-1ØØ) permite escoger una nota de forma aleatoria; la línea 7Ø calcula su nombre, y la línea 8Ø dibuja la nota.
El nombre de la nota se visualiza en la línea 9Ø.
La línea 11Ø permite volver a empezar.

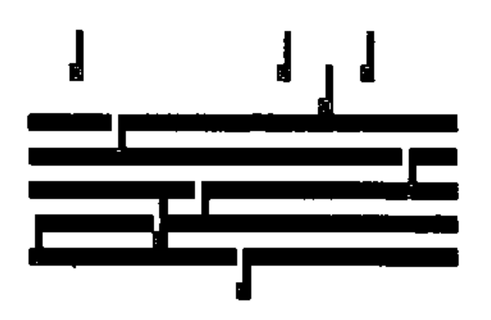
PARA JUGAR: Poner en marcha la ejecución del programa pulsando RUN.
Entonces se visualizará rápidamente el pentagrama, después se representan las notas, con su nombre visualizado en la parte inferior de la pantalla.
Una presión en cualquier tecla provocará la visualización de una nueva serie de notas, y una presión en la tecla BREAK provocará el final de la partida.

POSIBLES EXTENSIONES: Podrán afinarse las reglas de composición, visualizar más notas, introducir sostenidos y bemoles, mejorar el grafismo (sobre todo, en la versión para el ZX Spectrum), hacer sonar la melodía visualizada (esto se realiza en la versión Spectrum, pero las notas no corresponden realmente a las notas visualizadas: deberán volver a calcularse los intervalos).

COMPONER

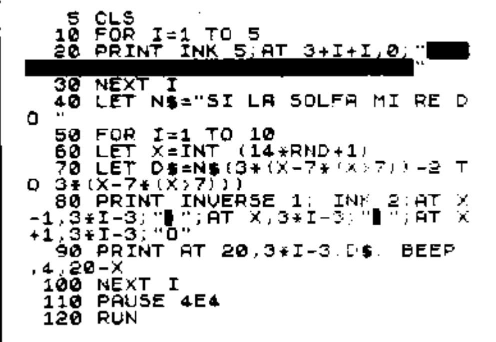
PROGRAMA ZX 81

```
   5 CLS
  10 FOR I=SGN PI TO 5
  20 PRINT AT INT PI+I+I,NOT PI;
"████████████████████████████████
"
  30 NEXT I
  40 LET N$="SI LA SOLFA MI RE D
O "
  50 FOR I=SGN PI TO 10
  60 LET X=INT (14*RND+SGN PI)
  70 LET D$=N$(INT PI*(X-7*(X>7)
)-2 TO INT PI*(X-7*(X>7)))
  80 PRINT AT X-SGN PI,INT PI*I-
INT PI;"▌";AT X,INT PI*I-INT PI;
"▌";AT X+SGN PI,INT PI*I-INT PI;
"█"
  90 PRINT AT 20,INT PI*I-INT PI
;D$
 100 NEXT I
 110 PAUSE 4E4
 120 RUN
```

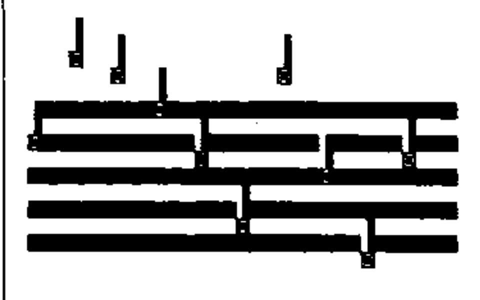


MI SI RE FA SOLDO SI SOLSI SI

PROGRAMA ZX SPECTRUM



```
   5 CLS
  10 FOR I=1 TO 5
  20 PRINT INK 5;AT 3+I+I,0;"█████
███████████████████████████"
  30 NEXT I
  40 LET N$="SI LA SOLFA MI RE D
O "
  50 FOR I=1 TO 10
  60 LET X=INT (14*RND+1)
  70 LET D$=N$(3*(X-7*(X>7))-2 T
O 3*(X-7*(X>7)))
  80 PRINT INVERSE 1; INK 2;AT X
-1,3*I-3;"▌";AT X,3*I-3;"▌";AT X
+1,3*I-3;"O"
  90 PRINT AT 20,3*I-3;D$: BEEP
.4,20-X
 100 NEXT I
 110 PAUSE 4E4
 120 RUN
```



RE SI LA FA DO FA LA SI RE DO

Nicomaque

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: juego de reflexión (cálculo mental).
Dificultad del juego: mediana.

EL JUEGO: Utiliza una propiedad algo particular de los enteros del 1 al 100, que pueden identificarse, sin ambigüedad, por sus restos en las divisiones por 3, 5 y 7.

El programa que aquí se propone es un programa codificador que simplemente visualiza los restos de un entero N dado en las divisiones indicadas.

El juego consiste en intentar descubrir el entero (pero, en esta versión, sólo puede efectuarse un intento).

EL PROGRAMA: La línea 1Ø permite confeccionar el entero N.

El bucle I (2Ø-7Ø) calcula los tres restos sucesivos, y el bucle intercalado J (4Ø-6Ø) permite visualizarlos en el emplazamiento apropiado, bajo una forma bastante visual.

La línea 8Ø está para indicar de qué división se trata.

El número propuesto por el jugador se introduce en la línea 1ØØ; se visualiza por la línea 1Ø5 y se comprueba en una variable lógica R$, que permitirá después la visualización del final de la partida (BRAVO o HA PERDIDO).

La línea 12Ø realiza la visualización del final.

Las líneas 13Ø a 15Ø se ocupan, según la respuesta del jugador, de la gestión de la siguiente partida.

PARA JUGAR: Se pone en marcha el programa pulsando RUN.

Se visualizarán en la misma línea los tres restos, simbolizados por puntos (la ausencia de puntos significa: resto 0).

A la pregunta SU PRUEBA, se responderá dando un entero comprendido entre 0 y 100, seguido de NEWLINE.

Se obtendrá en respuesta una visualización HA PERDIDO o BRAVO, y después la pregunta OTRA PARTIDA ¿(S/N)?

Se responderá 0 seguido de NEWLINE para hacer una nueva prueba.

POSIBLES EXTENSIONES: Puede considerarse la realización del programa descifrador (se dan los restos y el ZX deberá presentar N) y buscar si puede ampliarse la propiedad (efectuando, por ejemplo, más divisiones).

NICOMAQUE

PROGRAMA ZX 81

```
   5 CLS
  10 LET X=INT (100*RND+SGN PI)
  20 FOR I=SGN PI TO INT PI
  30 LET A=X-(I+I+SGN PI)*INT (X
/(I+I+SGN PI))
  40 FOR J=SGN PI TO A
  50 PLOT 15*I+2*J+4,28
  60 NEXT J
  70 NEXT I
  80 PRINT AT 10,NOT PI;"RESTO";
TAB 9;"POR 3";TAB 17;"POR 5";TAB
 25;"POR 7"
  90 PRINT AT 15,NOT PI;"SU PRUE
BA :";
 100 INPUT N
 105 PRINT N
 110 LET R$=("BRAVO" AND N=X)+("
HA PERDIDO" AND N<>X)
 120 PRINT R$
 130 PRINT "OTRA PARTIDA (S/N) ?
"
 140 INPUT R$
 150 IF R$="S" THEN RUN
```

```
RESTO     POR 3    POR 5    POR 7


SU PRUEBA :22
BRAVO
OTRA PARTIDA (S/N) ?
```

PROGRAMA ZX SPECTRUM

```
   5 CLS : BORDER 4
  10 LET X=INT (100*RND+1)
  20 FOR I=1 TO 3
  30 LET A=X-(I+I+1)*INT (X/(I+I
+1))
  40 FOR J=1 TO A
  50 INK 1: CIRCLE 60*I+5*J+15,8
0,3
  60 NEXT J
  70 NEXT I
  80 PRINT INVERSE 1; INK 2;AT 1
0,0;"RESTO";TAB 9;"POR 3";TAB 17
;"POR 5";TAB 25;"POR 7"
  90 PRINT AT 15,0;"SU PRUEBA :"
;
 100 INPUT N
 105 PRINT N
 110 LET R$=("BRAVO" AND N=X)+("
HA PERDIDO" AND N<>X)
 120 PRINT FLASH 1; BRIGHT 1;R$:
 BEEP 1,10
 130 PRINT "OTRA PARTIDA (S/N) ?
"
 140 INPUT R$
 150 IF R$="S" THEN RUN
```



```
SU PRUEBA :23
HA PERDIDO
OTRA PARTIDA (S/N) ?
```

Permutaciones

REFERENCIAS:

Modalidad: SLOW o FAST.

Dificultad de programación: mediana.

Categoría del juego: de reflexión.

Dificultad del juego: mediana.

EL JUEGO:

El ZX solicita primero el número de letras que se quieren permutar (no más de 12). A partir de aquí, va a visualizar el número correspondiente de letras, tomadas del principio del alfabeto, primero en desorden (PRINCIPIO), y después en orden (FINAL).

Debe lograrse ordenar las letras dadas al principio utilizando las permutaciones sobre las n primeras letras a partir de la izquierda, con el menor número de intentos posible.

EL PROGRAMA:

La línea 3Ø permite escoger el número de letras, y la línea 4Ø controla que no se pidan más de 12.

La línea 6Ø realiza la mezcla, escogiendo al azar la nueva posición de cada letra.

A continuación, las letras se incorporan en una variable B$ (línea 7Ø); a medida que van saliendo, se suprimen en A$ y se vuelven a poner en orden en C$ (líneas 75 y 8Ø).

La línea 9Ø controla el número de letras de B$.

Las visualizaciones se realizan en las líneas 1ØØ y 115.

La inversión se hace letra a letra, en el bucle I (14Ø-16Ø); el resultado se sitúa en B$, en la línea 17Ø, se visualiza por la línea 18Ø y se compara con el resultado final en la línea 19Ø.

El número de pruebas se visualiza al final de la partida, gracias a la línea 2ØØ.

PARA JUGAR:

Después de pulsar RUN, se obtiene la visualización de la pregunta NÚM. DE LETRAS (DE 1 A 12); se responde pulsando N, seguido de NEWLINE.

La visualización siguiente presenta el orden inicial (PRINCIPIO) y el que debe obtenerse (FINAL).

Se juega indicando el número de letras que tienen que permutarse a partir de la izquierda, por ejemplo: en una visualización inicial de GEABDFC, si se pulsa 3 y NEWLINE, se obtendrá AEGBDFC.

Se continuará hasta obtener el orden deseado.

La puntuación se visualizará al final de la partida, lo mismo que la pregunta OTRA PARTIDA (S o N); se contestará S si quiere realizarse otra partida.

POSIBLES EXTENSIONES: **Podrá utilizarse, por ejemplo, el mismo tipo de juego para reconstruir un dibujo escondido, cuyas distintas partes serían localizadas por letras. También pueden realizarse permutaciones más complejas (por ejemplo, el hecho de permutar dos letras provocaría la permutación de las dos letras más próximas de la que está más a la izquierda, etc.).**

PERMUTACIONES

PROGRAMA ZX 81

```
   5 CLS
  10 LET A$="ABCDEFGHIJKL"
  20 PRINT "NUMERO DE LETRAS (DE
 1 A 12)"
  30 INPUT N
  35 LET C$="            "
  40 IF N>LEN C$ THEN RUN
  50 LET B$=""
  60 LET X=INT (RND*N+SGN PI)
  65 IF A$(X)=" " THEN GO TO COD
E "U"
  70 LET B$=A$(X)+B$
  75 LET C$(X)=A$(X)
  80 LET A$(X)=" "
  85 LET S=NOT PI
  90 IF LEN B$<>N THEN GO TO COD
E "U"
  95 LET C$=C$( TO LEN B$)
 100 PRINT "PRINCIPIO :";B$,," FI
NAL :";C$
 110 LET S=S+SGN PI
 115 PRINT AT 7,SGN PI;"PERMUTAC
ION :";S
 120 INPUT L
 130 LET A$=""
 140 FOR I=SGN PI TO L
 150 LET A$=A$+B$(L-I+SGN PI)
 160 NEXT I
 170 LET B$=A$+B$(L+SGN PI TO
 180 PRINT "RESULTADO :";B$
 190 IF B$<>C$ THEN GO TO 105
 200 PRINT "ENCONTRADO EN ";S," 
PRUEBAS"
 210 PRINT "OTRA PARTIDA (S O N)
?"
 220 INPUT A$
 230 IF A$="S" THEN RUN
```

```
NUMERO DE LETRAS (DE 1 A 12)
PRINCIPIO :EBCFAD
  FINAL :ABCDEF


  PERMUTACION :9
RESULTADO :ABCDEF
ENCONTRADO EN 9 PRUEBAS
OTRA PARTIDA (S O N)?
```

PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 LET A$="ABCDEFGHIJKL"
  20 PRINT "NUMERO DE LETRAS (DE
 1 A 12) ";
  30 INPUT N: PRINT N
  40 IF N>12 THEN RUN
  50 LET C$="            "
  55 LET B$=""
  60 LET X=INT (RND*N+1)
  65 IF A$(X)=" " THEN GO TO 60
  70 LET B$=A$(X)+B$
  75 LET C$(X)=A$(X)
  80 LET A$(X)=" "
  85 LET S=1
  90 IF LEN B$<>N THEN GO TO 60
  95 LET C$=C$( TO LEN B$)
 100 PRINT ;"PRINCIPIO :"; PAPER
 4;B$: PRINT "FINAL     :"; PAPE
R 5;C$
 105 PRINT AT 4,0;"RESULTADO ";S
-1;" :";B$
 112 BEEP 1,10: PAUSE 5: BEEP 1,
10: PAUSE 5: BEEP 1,10: PRINT AT
 12,10;"                 "
 115 PRINT AT 7,1;"PERMUTACION :
"; FLASH 1; INK 2;S
 120 INPUT L
 125 LET S=S+1
 130 LET A$=""
 140 FOR I=1 TO L
 150 LET A$=A$+B$(L-I+1)
 160 NEXT I
 170 LET B$=A$+B$(L+1 TO )
 180 PRINT INK 2;AT 12,0;"RESULT
ADO :"; PAPER 7; INK 1; FLASH 1;
B$
 190 IF B$<>C$ THEN GO TO 105
 200 PRINT "ENCONTRADO EN ";S-1;
" PRUEBAS"
 210 PRINT "OTRA PARTIDA (S O N)
?"
 220 INPUT A$
 230 IF A$="S" THEN RUN
```

```
NUMERO DE LETRAS (DE 1 A 12)
PRINCIPIO :EBCFAD
  FINAL :ABCDEF


  PERMUTACION :9
RESULTADO :ABCDEF
ENCONTRADO EN 9 PRUEBAS
OTRA PARTIDA (S O N)?
```

```
NUMERO DE LETRAS (DE 1 A 12) 6
PRINCIPIO :AEFDBC
FINAL     :ABCDEF

RESULTADO 2 :BCDFEA


  PERMUTACION :2


 RESULTADO :BCDFEA
```

Juego de cartas

REFERENCIAS:

Modalidad: FAST o SLOW.
Dificultad de programación: difícil.
Categoría del juego: juego de prestidigitación.
Dificultad del juego: fácil.

EL JUEGO:

Es un juego bastante clásico, pero aquí es el ZX el que manipula las cartas y el que realiza la función de prestidigitador; se visualizan 21 cartas tomadas al azar de un paquete de 52, y el jugador deberá retener mentalmente una de las cartas presentadas. Después, deberá indicar al ZX en qué columna (1.ª, 2.ª o 3.ª, a partir de la izquierda) se encuentra la carta escogida; las cartas vuelven a distribuirse y el jugador indica nuevamente la columna correspondiente a su carta; esto volverá a repetirse una vez más, y a la 4.ª visualización, el ZX estará en condiciones de adivinar la carta escogida (en la versión del ZX 81 no se designa la carta debido a la falta de memoria, lo que permitirá asombrar fácilmente a las amistades, siempre que se recuerde que la carta se encuentra en el 4.° lugar de la 2.ª columna).

EL PROGRAMA:

El bucle I (2Ø-8Ø) permite escoger al azar 21 cartas entre 52, y el bucle intercalado J (4Ø-6Ø) permite asegurar que no se efectúa dos veces la misma extracción.

La línea 1ØØ contiene los colores de las cartas, así como los nombres de las figuras (V, D, R y A), en una variable alfanumérica B$.

El bucle I (11Ø-16Ø) realiza la visualización del juego.

La línea 12Ø descifra el color de la carta; la línea 13Ø escoge la posición de la visualización, y la línea 14Ø descifra la altura.

La visualización propiamente dicha se efectúa en la línea 15Ø.

La línea 17Ø permite al jugador introducir el número de la columna en la que se encuentra la carta. El bucle J (185-195) realiza la mezcla, y los bucles intercalados J (2ØØ-24Ø) e I (21Ø-23Ø) confeccionan la nueva variable A$, que tendrá que utilizarse en la siguiente visualización.

PARA JUGAR:

Se pone en marcha el programa con RUN.

Se obtiene una primera visualización de 21 cartas, repartidas en 3 columnas.

Entonces, se indica el número (1, 2 o 3) de la columna donde se encuentra la carta escogida, seguido de NEWLINE.

Se obtiene una segunda visualización y se hace lo mismo, y después una tercera visualización, repitiendo la misma operación.

A la cuarta visualización, la carta escogida se encuentra en la 4.ª posición de la 2.ª columna (el ZX 81, en su versión de 1K, no permite indicarla, pero sí se indica en la versión Spectrum, más completa).

Se saldrá del programa pulsando STOP seguido de NEWLINE, en lugar del número de la columna.

POSIBLES EXTENSIONES: Se considerará una mejora en la visualización escribiendo los nombres completos de las cartas o, también, dibujando verdaderas cartas en color.

Asimismo, pueden probarse otros juegos utilizando el mismo principio de base, pero con más cartas, etc.

JUEGO DE CARTAS

PROGRAMA ZX 81

```
  10 LET A$=""
  12 LET R=VAL "2"
  15 LET Q=VAL "7"
  20 FOR I=SGN PI TO INT PI*Q
  30 LET B$=CHR$ INT (CODE "£"+C
ODE "O"*RND)
  40 FOR J=SGN PI TO LEN A$
  50 IF B$=A$(J) THEN GO TO CODE
 "2"
  60 NEXT J
  70 LET A$=A$+B$
  80 NEXT I
 100 LET B$="TRDICOPIVDRA"
 110 FOR I=SGN PI TO LEN A$
 120 LET S=INT ((CODE A$(I)-CODE
 "£")/CODE "$")
 130 LET J=Q*INT ((I-SGN PI)/Q)
 140 LET T=SGN PI+CODE A$(I)-COD
E "£"-CODE "$"*S
 150 PRINT AT I-J,J;(STR$ (T+SGN
 PI) AND T<=Q+R)+(B$(T) AND T>Q+
R);TAB J+INT PI;B$(R*S+SGN PI TO
 R*S+R)
 160 NEXT I
 170 INPUT N
 175 CLS
 180 LET R$="20120"
 182 LET B$=""
 185 FOR J=NOT PI TO R.
 190 LET B$=B$+A$(Q*VAL R$(N+J)+
SGN PI TO Q*VAL R$(N+J)+Q)
 195 NEXT J
 197 LET A$=""
 200 FOR J=NOT PI TO R
 210 FOR I=SGN PI TO Q
 220 LET A$=A$+B$(INT PI*I-R+J)
 230 NEXT I
 240 NEXT J
 250 GO TO CODE "COS "/R
```

```
VA DI  7  TR  2  DI
RE PI  9  PI  3  PI
RE CO  4  CO  5  PI
6  CO  5  CO  RE DI
8  TR  7  DI  DA PI
VA PI  AS TR  4  PI
5  DI  VA CO  RE TR
```

PROGRAMA ZX SPECTRUM

```
  10 LET A$=""
  12 LET R=2
  15 LET Q=7
  20 FOR I=1 TO 21
  30 LET B$=CHR$ INT (38+52*RND)
  40 FOR J=1 TO LEN A$
  50 IF B$=A$(J) THEN GO TO 30
  60 NEXT J
  70 LET A$=A$+B$
  80 NEXT I
  90 LET H=3
  95 LET P=1
 100 LET B$="TRDICOPIVADAREAS"
 110 FOR I=1 TO LEN A$*P+11*NOT
P
 120 LET S=INT ((CODE A$(I)-38)/
13)
 130 LET J=Q*INT ((I-1)/Q)
 140 LET T=CODE A$(I)-37-13*S
 145 IF NOT P THEN PRINT INVERSE
 1;AT 0,0;"SU CARTA ES  ", INVER
SE 0; PAPER 2;"      "
 150 PRINT AT (I-J)*P,J*P+18*NOT
 P;(STR$ (T+1) AND T<=Q+R)+(B$(A
BS (2*T-11) TO ABS (2*T-10)) AND
 T>Q+R);AT (I-J)*P,(J+3)*P+21*NO
T P;B$(R*S+1 TO R*S+R)
 160 NEXT I
 165 IF NOT P THEN STOP
 170 INPUT "QUE COLUMNA :";N
 175 CLS
 177 LET H=H-1
 180 LET R$="20120"
 182 LET B$=""
 185 FOR J=0 TO R
 190 LET B$=B$+A$(Q*VAL R$(N+J)+
1 TO Q*VAL R$(N+J)+Q)
 195 NEXT J
 196 IF NOT H THEN LET A$=B$: LE
T P=0: GO TO 100
 197 LET A$=""
 200 FOR J=0 TO R
 210 FOR I=1 TO Q
 220 LET A$=A$+B$(3*I-R+J)
 230 NEXT I
 240 NEXT J
 250 GO TO 100
```

```
5  DI  10 DI  AS DI
9  DI  8  CO  6  PI
2  DI  5  TR  3  TR
2  TR  AS PI  DA DI
4  TR  9  CO  10 PI
4  DI  DA PI  5  PI
10 CO  4  PI  7  CO
```

NIVEL 4

N.°	*Nombre del programa*	*Tema y particularidades*
1	**Tirada de dados**	Suficientemente flexible para utilizarse en todas partes.
2	**Juego de la lotería**	Ganar en la siguiente tirada, sin problemas.
3	**La bomba**	¿Sabrá encontrarla a tiempo? (para un jugador).
4	**Pizarra mágica**	Para poner a punto sus creaciones gráficas, sin papel ni lápiz.
5	**Anuncio publicitario**	Utilizar las posibilidades de tratamiento de cadenas del ZX.
6	**Evasión**	Hará falta mucha suerte para tener éxito.
7	**Sombreros**	Sea muy observador y tendrá éxito (para dos jugadores).
8	**Combate aéreo**	¡No es tan fácil derribar un aparato que se mueve todo el tiempo!
9	**Dibujo de dados**	Para mejorar los juegos de dados incorporando este programa.
10	**El juego del misil**	¡Una misión muy peligrosa! (para un solo jugador).
11	**El mono**	Un juego de letras para dos jugadores, en el que hacen falta buenos reflejos.
12	**El juego del 15**	Un juego de reflexión, en el que se utilizan cifras (para dos jugadores).
13	**Destreza**	Juego de reflexión muy visual, para un jugador.
14	**Submarino**	Un clásico de los videojuegos (para un jugador).
15	**Zombis**	Un juego extravagante, pero muy visual (para un jugador).
16	**Diago**	¿Está realmente seguro de sus reflejos? (para un jugador).
17	**Topología**	Una dirección a explorar, para otros juegos (uno o dos jugadores).
18	**Dar caza al submarino**	Harán falta muchos reflejos y mucha suerte.
19	**El lagarto de Alicia**	Mucha suerte, aunque muy espectacular (para un jugador).
20	**Calculadora**	Un programa de simulación que podrá fácilmente ser mejorado.

Animación

REGISTRO DE CARACTERES TECLEADOS : INKEY$

El principio de un juego gráfico, en animación, se basa en que debe desarrollarse a solas, de una cierta manera y durante un determinado tiempo, mientras el jugador no haya realizado ninguna intervención.

Debe, pues, disponerse de la posibilidad de detectar si ha existido intervención del jugador o no, sin por ello interrumpir el desarrollo del programa.

En el Basic Sinclair, se ofrece esta propiedad por medio de la **función INKEY$,** que realiza una exploración de todas las teclas y graba el nombre de la última tecla pulsada.

Podrán utilizarse INKEY$ de dos maneras:

- en un bucle (realización de un movimiento), se comprueban los INKEY$ para salir, llegado el caso, del bucle;
- mientras INKEY$ está vacío (no se ha pulsado nada) se espera y se sale cuando se pulse algo.

Eventualmente, podrán producirse problemas de rechazo (principalmente en el ZX 81 al descifrar el carácter cogido por INKEY$.

DESPLAZAMIENTO DE OBJETOS DE ABAJO ARRIBA DE LA PANTALLA: SCROLL

Si se trabaja en Basic, es difícil gestionar el desplazamiento simultáneo de varios objetos sobre la pantalla (en caso contrario, se obtiene rápidamente una animación irregular, debida a un tiempo de desplazamiento prohibitivo).

El Basic del ZX 81 dispone de una instrucción que permite poner remedio a esto y que puede utilizarse en numerosos juegos: se trata de la **instrucción SCROLL,** que simplemente hace subir una línea, hacia arriba, los diversos objetos (textos gráficos) que están presentes en la pantalla en el momento de su ejecución. Evidentemente, la línea de arriba se pierde. Esto permite simular la subida hacia arriba de la pantalla de un conjunto de objetos, con borrado automático de la posición anterior. Por supuesto, un objeto que debería permanecer inmóvil al emitirse un SCROLL, deberá dibujarse de nuevo permanentemente en su buena posición (y borrado de la mala, salvo si está en lo alto de la pantalla).

El ZX Spectrum no dispone de una instrucción SCROLL, aunque lo realiza a petición cuando la pantalla está llena; podrá aprovecharse esta propiedad reemplazando el SCROLL del ZX 81 por una secuencia adaptada del juego considerado.

Tirada de dados

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: ninguna.
Categoría del juego: tirada de dados.
Dificultad del juego: ninguna.

EL JUEGO: El programa que se propone no realiza un juego por sí mismo, pero podrá utilizarse como subprograma para numerosos juegos.

Ofrece simplemente la posibilidad de tirar de 1 a 9 dados por simple presión de una tecla. Los dados se visualizan después, con un grafismo muy simplificado.

EL PROGRAMA: La línea 2Ø capta el carácter pulsado en el teclado y lo transforma en número (si no es posible, el programa se para con un mensaje de error).

Una variable D se dimensiona en la línea 35, para recibir el valor indicado para cada dado.

Un bucle J (43-45) se utiliza para dibujar un fondo gris, y otro bucle I (4Ø-8Ø) sirve para la tirada (tirada en la línea 5Ø e impresión en la línea 7Ø).

Las líneas 9Ø a 11Ø ofrecen la posibilidad de volver a tirar; estas líneas no son indispensables.

PARA JUGAR: Se pone en marcha el programa por la orden RUN y se obtiene la visualización NÚMERO DE DADOS PARA TIRAR, al que se responde tecleando una cifra.

Entonces el resultado se visualiza automáticamente; bastará pulsar cualquier tecla para volver a empezar el programa.

Para parar necesariamente se pulsará BREAK.

POSIBLES EXTENSIONES: Podrá utilizarse el boceto proporcionado por este programa para la realización de todos los juegos en los que se precisen dados (421, YAM'S, CRAPS, etc.).

TIRADA DE DADOS

PROGRAMA ZX 81

```
 10 PRINT "NUMERO DE DADOS PARA
TIRAR"
 15 PAUSE 4E4
 20 LET N=VAL INKEY$
 30 CLS
 35 DIM D(N)
 40 FOR I=SGN PI TO N
 43 FOR J=SGN PI TO INT PI
 44 PRINT AT 8+J,3*I-1;"█";AT 8
+J,3*I;"█";AT 8+J,3*I+1;"█"
 45 NEXT J
 50 LET D(I)=INT (RND*6)+SGN PI
 70 PRINT AT 10,3*I;D(I)
 80 NEXT I
 90 PAUSE 4E4
100 CLS
110 GO TO 10
```



PROGRAMA ZX SPECTRUM

```
  5 GO TO 100
 10 PRINT AT 0,0;"NUMERO DE DAD
OS PARA TIRAR :"
 15 PAUSE 4E4
 20 LET N=VAL INKEY$
 30 PRINT AT 0,28;N
 35 DIM D(N)
 40 FOR I=1 TO N
 43 FOR J=1 TO 3
 44 PRINT INK 2;AT 8+J,3*I-1;"█
";AT 8+J,3*I;"█";AT 8+J,3*I+1;"█
"
 45 NEXT J
 50 LET D(I)=INT (RND*6)+1
 70 PRINT INK 2; INVERSE 1;AT 1
0,3*I;D(I)
 80 NEXT I
 90 BEEP .3,10: BEEP .3,18: BEE
P .3,8
100 PAUSE 4E4: PAPER 5: CLS
110 GO TO 10
```

NUMERO DE DADOS PARA TIRAR :6



El juego de la lotería

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: fácil.
Categoría del juego: de azar (para varios jugadores).
Dificultad del juego: fácil.

EL JUEGO: Se trata de realizar una tirada completa de lotería; es decir, hay que obtener 6 números, comprendidos entre el 1 y el 49, más un número complementario, todos distintos, que podrían, por ejemplo, utilizarse para llenar las casillas de un juego de lotería.

EL PROGRAMA: Se utiliza una tabla N (7), dimensionada en la línea 1Ø, para grabar los resultados.

La tirada de los números se efectúa en un bucle I (2Ø-7Ø); un segundo bucle J (4Ø-6Ø) se utiliza para comprobar que no se realicen dos tiradas idénticas.

La tirada (o la nueva tirada) se efectúa en la línea 5Ø.

El bucle I (9Ø-11Ø) visualiza el resultado, y la línea 12Ø el número complementario.

Las líneas 13Ø a 15Ø se ocupan del caso, cuando se quiera volver a realizar otra partida.

PARA JUGAR: Basta con pulsar RUN para poner en marcha el programa y obtener la visualización de una primera lotería.

La pregunta ¿QUIERE OTRA TIRADA (S o N)? se visualiza en la parte inferior de la pantalla; se responderá S si se desea continuar.

POSIBLES EXTENSIONES: Puede imaginarse un programa que podría tener en cuenta sus propios resultados o los resultados introducidos por un jugador, con el fin de mejorar la tirada, teniendo en cuenta las probabilidades de sacar un número dado.

EL JUEGO DE LA LOTERÍA

PROGRAMA ZX 81

```
  5 CLS
 10 DIM N(7)
 20 FOR I=1 TO 7
 30 LET N(I)=INT (RND*49+1)
 40 FOR J=1 TO I-1
 50 IF N(I)=N(J) THEN GO TO 30
 60 NEXT J
 70 NEXT I
 80 PRINT AT 3,5;"TIRADA DE LOT
ERIA"
 90 FOR I=1 TO 6
100 PRINT AT 5,4*I+3;N(I)
110 NEXT I
120 PRINT AT 8,1;"NUMERO COMPLE
MENTARIO :";N(7)
130 PRINT AT 12,1;"QUIERE OTRA
PARTIDA (S O N)?"
140 PAUSE 4E4
150 IF INKEY$="S" THEN RUN
```

```
     TIRADA DE LOTERIA

       36  20  25  26  17  45

NUMERO COMPLEMENTARIO :39


QUIERE OTRA PARTIDA (S O N)?
```

PROGRAMA ZX SPECTRUM

```
  5 CLS
 10 DIM N(7)
 20 FOR I=1 TO 7
 30 LET N(I)=INT (RND*49+1)
 40 FOR J=1 TO I-1
 50 IF N(I)=N(J) THEN GO TO 30
 60 NEXT J
 70 NEXT I
 80 PRINT INK 2; FLASH 1;AT 3,7
;"TIRADA DE LOTERIA"
 90 FOR I=1 TO 6
100 PRINT PAPER 6; INK 1, BRIGH
T 1;AT 8,4*I+3;N(I)
110 NEXT I
120 PRINT AT 10,1;"NUMERO COMPL
EMENTARIO  : "; INK 3; FLASH 1;N
(7): BEEP .4,10: BEEP .4,15: BEE
P .4,10
130 PRINT INVERSE 1;AT 15,0;"QU
IERE OTRA PARTIDA (S O N)?"
140 PAUSE 4E4
150 IF INKEY$="S" THEN RUN
```

```
      TIRADA DE LOTERIA


      47  48  3   46  44  49

 NUMERO COMPLEMENTARIO  : 41
```



La bomba

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: mediana.
Categoría del juego: solitario.
Dificultad del juego: mediana.

EL JUEGO: Debe encontrarse, con la ayuda de un detector especializado, un paquete radiactivo con trampa que un viajero con malas intenciones ha colocado en una habitación de un hotel. El único problema, pero muy importante, es que el hotel tiene un millón de habitaciones (100 * 100 * 100).

El detector visualiza permanentemente una señal que informa sobre la proximidad de la bomba y el tiempo que queda, descontado de 10 en 10.

¿Podrá descubrirse la bomba antes de que haga explosión?

EL PROGRAMA: Una variable numérica R se dimensiona en 3, en la línea 3Ø, a fin de contener las tres coordenadas que definen la posición del jugador.

La posición de la habitación en la que se encuentra el paquete, se define por un cálculo aleatorio de tres variables A, B, C, en las líneas 4Ø, 5Ø, 52.

El bucle I (55-62) dibuja el detector.

La señal de aproximación se calcula en la línea 63, y la línea 9Ø permite la visualización de las coordenadas de la habitación que debe descubrirse, en el caso que se haya perdido.

La línea 1ØØ permite la visualización simultánea de la señal y del tiempo que queda por transcurrir.

La línea 11Ø realiza el descuento del tiempo.

La línea 12Ø controla si se ha encontrado o no.

El bucle I (14Ø-18Ø) permite introducir las nuevas coordenadas, utilizando el subprograma situado en las líneas 2ØØ a 21Ø.

PARA JUGAR: Después de la puesta en marcha del programa con RUN, se visualizan la señal y el tiempo (en principio 200).

Entonces, el jugador se encuentra en la posición x = 0, y = 0, z = 0, y deberán introducirse sucesivamente las tres coordenadas *x, y, z* de la habitación que debe explorarse.

Por ejemplo, se pulsará 13, después NEWLINE, 45, después NEWLINE, etc.

Se visualiza entonces la nueva señal, y se reduce el tiempo en 10.

Se continuará así hasta dar con el resultado (parada del juego y final del programa) o hasta la aparición de tres números (x, y, z) sin salida del programa, lo que significa que se ha perdido.

POSIBLES EXTENSIONES: Será indispensable, sin duda, mejorar el grafismo, añadir una explosión (sonora y visual) para simbolizar la explosión, visualizar simbólicamente la posición del hotel, etc.

LA BOMBA

PROGRAMA ZX 81

```
  20 LET L=100
  25 LET T=L+L
  30 DIM R(INT PI)
  40 LET A=INT (RND*L)
  50 LET B=INT (RND*L)
  52 LET C=INT (RND*L)
  55 FOR I=NOT PI TO CODE ")"
  60 PRINT AT 4,I;"█";AT 5,I;"█"
;AT 6,I;"█"
  62 NEXT I
  63 LET S=ABS ((A/L+B+C*L)-(R(S
GN PI)/L+R(2)+R(INT PI)*L))
  90 IF NOT T THEN PRINT A,B,C
  95 PRINT AT 5,15;"  "
 100 PRINT AT 5,INT PI,S;TAB 13;
T
 110 LET T=T-INT SQR L
 120 IF NOT S THEN STOP
 130 PRINT AT SQR L,INT PI;"- X
";AT 12,INT PI;"- Y ";AT 14,INT
PI;"- Z "
 140 FOR I=SGN PI TO INT PI
 160 LET R$=""
 170 GOSUB CODE "COS "
 175 LET R(I)=VAL R$
 180 NEXT I
 190 GOTO CODE "Z"
 200 PRINT AT 8+2*I,7;"?";R$;"
"
 205 PAUSE 4E4
 210 IF CODE INKEY$=118 THEN RET
URN
 220 LET R$=R$+INKEY$
 240 GOTO CODE "COS "
```

```
[illegible]   [illegible]

- X ?

- Y ?76

- Z ?88
```

PROGRAMA ZX SPECTRUM

```
  20 LET L=100
  25 LET T=L+L
  30 DIM R(3)
  40 LET A=INT (RND*L)
  50 LET B=INT (RND*L)
  52 LET C=INT (RND*L)
  55 FOR I=0 TO 17
  60 PRINT INK 5;AT 4,I;"█";AT 5
,I;"█";AT 6,I;"█"
  62 NEXT I
  63 LET S=ABS ((A/L+B+C*L)-(R(1
)/L+R(2)+R(3)*L))
  90 IF NOT T THEN PRINT INK 4;
FLASH 1;AT 20,0;"HA PERDIDO  ";A
;"  ";B;"  ";C;"  ": BEEP .3,0:
BEEP .3,0: BEEP .3,0
  95 PRINT AT 5,15;"  "
 100 PRINT AT 5,3;S;AT 5,13; FLA
SH 1;T
 110 LET T=T-INT SQR L: BEEP 1,1
5
 120 IF NOT S THEN PRINT AT 21,0
; PAPER 2; FLASH 1;"BRAVO"
 130 PRINT INK 2;AT SQR L,3;"- X
:       ";AT 12,3;"- Y :       ";AT
 14,3;"- Z :       "
 140 FOR I=1 TO 3
 160 LET R$=""
 170 GO SUB 200
 175 LET R(I)=VAL R$
 180 NEXT I
 190 GO TO 63
 200 PRINT AT 8+2*I,8; INK 2; FL
ASH 1;"?"; INK 0; FLASH 0;" ";R$
;"  "
 205 PAUSE 4E4
 210 IF CODE INKEY$=13 THEN BEEP
 .3,10: RETURN
 220 LET R$=R$+INKEY$
 240 GO TO 200
```

```
1785.98   200

- X :? 12

- Y :? 34

- Z :? 53
```

Pizarra mágica

REFERENCIAS:

Modalidad: SLOW o FAST.

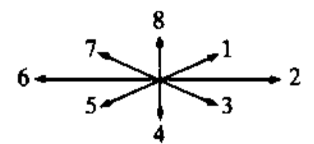
Dificultad de programación: fácil.

Categoría del juego: dibujo.

Dificultad del juego: fácil.

EL JUEGO:

Como el título parece indicar, se trata simplemente de utilizar la pantalla del televisor como una pizarra, manejando un punto en la pantalla que sigue 8 direcciones, con la ayuda de las teclas del 1 al 8, siguiendo las direcciones:



También se dispone de la posibilidad de desplazar el punto en la pantalla sin dejar rastro, pulsando la tecla Ø (esto también permitirá el borrado parcial), y podrá volver a incluirse pulsando 9.

Para borrar completamente un dibujo deberá pulsarse la tecla P, y para conservar un trazado en la impresora se saldrá del programa por BREAK antes de pulsar COPY.

EL PROGRAMA:

La línea 1ØØ realiza la visualización de un punto en la pantalla.

Las líneas 16Ø a 17Ø captan al vuelo la tecla que se ha pulsado (de 0 a 9, lo mismo que P).

La línea 25Ø llama a uno de los subprogramas que efectúan el desplazamiento.

Las líneas 3ØØ y 35Ø controlan que no se salga de la pantalla.

La línea 37Ø permite borrarlo todo, si se pulsa la tecla P.

PARA JUGAR:

Se pone en marcha el programa pulsando RUN; después puede desplazarse el punto que aparece abajo, a la izquierda de la pantalla, utilizando las 8 teclas de desplazamiento.

La tecla Ø permite desplazar el punto sin que deje ningún trazo (se borrará un trazo anterior si se pasa por encima del mismo).

La tecla 9 permite registrar el desplazamiento.

La tecla P lo borra todo, y vuelve a empezarse.

Para salir del programa se pulsará BREAK.

POSIBLES EXTENSIONES: Podrán considerarse posibilidades particulares para trazar rectas, círculos, dibujar con varios colores, mejorar la finura del trazo, etc. Si es necesario, podrá añadirse un programa que salve el dibujo obtenido, para una posterior utilización.

PIZARRA MÁGICA

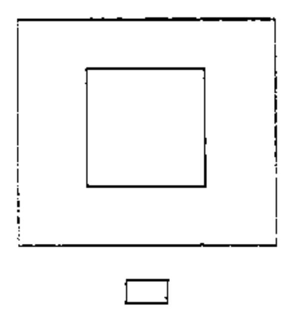
PROGRAMA ZX 81

```
   5 CLS
  10 LET X=10
  20 LET Y=X
  80 LET S=NOT PI
 100 PLOT X,Y
 150 PAUSE 4E4
 160 IF INKEY$="0" THEN GOTO 500
 165 IF INKEY$="9" THEN GOTO 80
 166 IF INKEY$="P" THEN RUN
 170 LET F=2*((VAL INKEY$)<5)-SG
N PI
 190 LET A=X
 200 LET B=Y
 250 GOSUB 1000*((VAL INKEY$)+2*
(F-SGN PI))
 300 LET C=X>NOT PI AND X<40 AND
 Y>NOT PI AND Y<30
 350 IF NOT C THEN GOSUB 5000
 370 IF S THEN UNPLOT A,B
 400 GOTO 100
 500 LET S=SGN PI
 510 GOTO 100
1000 LET Y=Y+F
2000 LET X=X+F
2010 RETURN
3000 GOSUB 2000
4000 LET Y=Y-F
4010 RETURN
5000 LET X=A
5010 LET Y=B
5020 RETURN
```



PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 LET X=10
  20 LET Y=X
  80 LET S=0
 100 PLOT X,Y
 150 PAUSE 4E4
 160 IF INKEY$="0" THEN GO TO 50
0
 165 IF INKEY$="9" THEN GO TO 80
 166 IF INKEY$="P" THEN RUN
 168 LET R=VAL INKEY$
 170 LET F=2*(R<5)-1
 190 LET A=X
 200 LET B=Y
 250 GO SUB 1000*(R+2*(F-1))
 300 LET C=X>0 AND X<255 AND Y>0
 AND Y<175
 350 IF NOT C THEN GO SUB 5000
 370 IF S THEN PLOT OVER 1;A,B
 400 GO TO 100
 500 LET S=1
 510 GO TO 100
1000 LET Y=Y+F
2000 LET X=X+F
2010 RETURN
3000 GO SUB 2000
4000 LET Y=Y-F
4010 RETURN
5000 LET X=A
5010 LET Y=B
5020 RETURN
```



Anuncio publicitario

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: mediana.
Categoría del juego: anuncio publicitario.
Dificultad del juego: ninguna.

EL JUEGO: Se introducirá un texto cualquiera al principio del programa. Podrá ser relativamente largo (4 o 5 líneas); a continuación, se escogerá el sentido y la velocidad de desfile, después de lo cual aparecerá el texto (una sola línea a la vez), que empezará a desfilar a la velocidad y en el sentido escogidos.

EL PROGRAMA: El texto se introduce en una variable P$, en la línea 3Ø.

Se truncará la visualización en 32 caracteres como máximo, gracias a la línea 13Ø.

El desfile a la izquierda se prepara en la línea 15Ø y el desfile a la derecha en la línea 16Ø.

El texto se visualizará en la línea 14Ø, y la duración de la visualización (velocidad de desfile) se escoge en la línea 17Ø.

La línea 175 permitirá cambiar, si es necesario, las condiciones de desfile.

PARA JUGAR: Se pone en marcha el programa con RUN, y ante la visualización de la pregunta SU TEXTO, se introducirá el texto que debe aparecer, seguido de NEWLINE.

Después se obtiene la visualización SENTIDO D o I, a la que se responderá tecleando D o I seguido de NEWLINE.

Después, a la pregunta VELOCIDAD, se responderá introduciendo un número pequeño para una velocidad rápida, y un número grande para una velocidad lenta (por ejemplo, de 0 a 40 o más).

El texto empieza a desfilar cuando se pulsa NEWLINE.

Si se quieren cambiar las condiciones de aparición del texto, basta con pulsar la tecla 0 para volver a encontrar las preguntas SENTIDO y VELOCIDAD.

Se pulsará BREAK para interrumpir el programa.

POSIBLES EXTENSIONES: Podrá introducirse color, realizar una visualización ondulante, hacer subir las líneas en la pantalla, etc.

ANUNCIO PUBLICITARIO

PROGRAMA ZX 81

```
  10 PRINT "INTRODUCIR EL TEXTO"
  20 PRINT
  30 INPUT P$
  35 LET P$=P$+" "
  40 PRINT "DESFILE D O I :";
  50 INPUT S$
  60 PRINT "DERECHA" AND S$(1)="
D";"IZQUIERDA" AND S$(1)="I"
  70 PRINT
  80 PRINT "VELOCIDAD DE 0 A 50"
  90 INPUT V
 105 CLS
 110 LET V=51-V
 120 LET T=LEN P$
 130 IF T>CODE "2" THEN LET T=CO
DE "2"
 140 PRINT AT INT PI,NOT PI;" "+
P$(SGN PI TO T)+" "
 150 IF S$(SGN PI)="I" THEN LET
P$=P$(2 TO LEN P$)+P$(SGN PI)
 160 IF S$(SGN PI)="D" THEN LET
P$=P$(LEN P$-SGN PI)+P$(SGN PI T
O LEN P$-SGN PI)
 170 PAUSE V*PI
 175 IF INKEY$="0" THEN GO TO 40
 180 GO TO 140
```

```
HASTA LA VISTA


, UN LOBITO BUENO  ERASE UNA V
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "INTRODUCIR EL TEXTO"
  20 PRINT
  30 INPUT P$: LET P$=" "+P$+" "
: LET A$=P$
  40 PRINT "DESFILE D O I :";
  50 INPUT S$
  60 PRINT "DERECHA" AND S$(1)="
D";"IZQUIERDA" AND S$(1)="I"
  70 PRINT
  80 PRINT "VELOCIDAD DE 0 A 50"
  90 INPUT V: LET V=51-V
 105 BORDER 4: PAPER 6: INK 2: C
LS
 120 LET T=LEN A$: LET P$=A$
 130 IF T>30 THEN LET T=30
 140 PRINT INK 2;AT 3,0;" "+P$(1
 TO T)+" "
 150 IF S$(1)="I" THEN LET P$=P$
(2 TO LEN P$)+P$(1)
 160 IF S$(1)="D" THEN LET P$=P$
(LEN P$-1)+P$(1 TO LEN P$-1)
 170 PAUSE V*PI
 175 IF INKEY$="0" THEN GO TO 40
 180 GO TO 140
```

```
ERASE UNA VEZ , UN LOBITO BUEN
```

Evasión

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: fácil.

Categoría del juego: solitario.

Dificultad del juego: fácil.

EL JUEGO: El único problema es lograr atravesar la pantalla sin ser tocado antes por uno de los numerosos obuses explosivos que dispara el ZX.

Se puede desplazar en 4 direcciones (con la ayuda de 4 flechas), y debe lograrse salir del terreno por el borde derecho sin haber sido tocado ni una sola vez.

EL PROGRAMA: Las líneas 5 y 7 definen la posición de salida del jugador.

La línea 1Ø escoge la coordenada horizontal de la casilla donde el ZX va a disparar (unida a la posición del jugador), y la línea 2Ø escoge la coordenada vertical (arbitraria) de esta casilla.

El desplazamiento del jugador está controlado por las líneas 7Ø y 8Ø; su trayectoria se traza por la línea 6Ø, y su posición actual (una H invertida en la pantalla) se imprime por la línea 9Ø.

La línea 1ØØ mira si se ha tocado o no.

La línea 11Ø efectúa el control del final de la partida.

PARA JUGAR: Se empieza la ejecución del programa por RUN.

Tiene lugar el primer disparo cuando aparece el jugador representado por una H invertida.

Las 4 teclas de dirección (5, 6, 7, 8) sirven para que el jugador se desplace.

Si es tocado, se obtiene en la pantalla el mensaje HA PERDIDO, y la partida termina.

Si logra salir por la derecha de la pantalla sin ser tocado, se obtiene el mensaje BRAVO, y la partida se acaba igualmente.

Puede acabarse en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: El color y un grafismo más cuidado mejorarán considerablemente la atracción del juego; también se podrá conseguir que el tiro de las bombas sea más aleatorio, introducir obstáculos que deban rodearse, añadir controles, hacer participar a varios jugadores, etc.

EVASIÓN

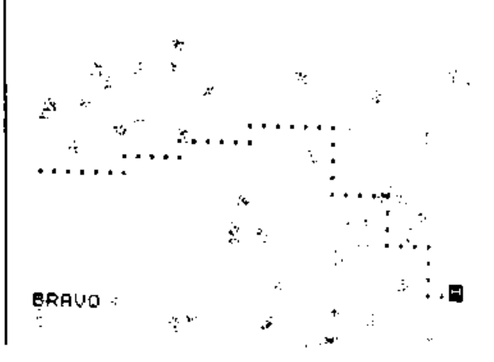
PROGRAMA ZX 81

```
   5 LET X=10
   7 LET Y=NOT PI
  10 LET H=2*(Y-1)+RND*5
  20 LET K=25*RND+5
  30 FOR J=1 TO 8*RND+4
  40 PLOT H+4*RND-2,K+4*RND-2
  50 NEXT J
  60 PRINT AT X,Y;" "
  70 LET X=X+(INKEY$="6")-(INKEY
$="7")
  80 LET Y=Y+(INKEY$="8")-(INKEY
$="5")
  90 PRINT AT X,Y;"█"
 100 IF ABS (H-2*Y)+ABS (K-2*(21
-X))<=4 THEN PRINT "HA PERDIDO";
U
 110 IF Y<33 THEN GOTO 10
```



PROGRAMA ZX SPECTRUM

```
   5 LET X=10
   7 LET Y=0
  10 LET H=8*(Y-1)+16*RND
  20 LET K=150*RND+5
  30 FOR J=1 TO 7*RND+3
  40 INK 2: PLOT H+6*RND-2,K+6*R
ND-2
  50 NEXT J
  60 INK 4: BEEP .3,-20: PRINT A
T X,Y;"."
  70 LET X=X+(INKEY$="6")-(INKEY
$="7")
  80 LET Y=Y+(INKEY$="8")-(INKEY
$="5")
  90 PRINT FLASH 1; INK 5;AT X,Y
;"█"
 100 IF ABS (H-8*Y)+ABS (K-8*(21
-X))<=12 THEN PRINT INK 0; FLASH
 1;"HA PERDIDO": BEEP .3,10: BEE
P .3,18: BEEP .3,8: STOP
 110 IF Y<30 THEN GO TO 10
 120 PRINT INK 1; FLASH 1;"BRAVO
": STOP
```



Sombreros

REFERENCIAS: Modalidad: SLOW o FAST.

Dificultad de programación: bastante fácil.

Categoría del juego: juego de reflejos, para dos jugadores.

Dificultad del juego: fácil.

EL JUEGO: La idea es muy simple: dos personajes se desplazan en la pantalla, uno hacia el otro, y se cruzan. En el momento en que se cruzan, es posible que se intercambien los sombreros.

Se trata de ser el primero en decir si ha habido intercambio de sombreros o no durante el cruce.

EL PROGRAMA: Las líneas 1Ø y 2Ø contienen el dibujo de los dos sombreros (en dos variables alfanuméricas B$ y C$).

La línea 3Ø contiene, en otra variable A$, los símbolos que permitirán dibujar los personajes.

Las líneas 6Ø a 8Ø permitirán el intercambio de los sombreros, si tiene lugar.

La variable T de la línea 9Ø sirve para comprobar si ha habido intercambio o no.

El dibujo de los dos personajes se realiza por un bucle I (13Ø-16Ø) y su desplazamiento por las líneas 17Ø y 18Ø.

La comprobación que permite el cambio de los sombreros o no se efectúa en la línea 11Ø.

Las líneas 19Ø a 21Ø determinan si el primer jugador que ha pulsado una tecla ha ganado o ha perdido.

PARA JUGAR: Poner en marcha el programa con RUN.

Inmediatamente, los personajes empiezan a desplazarse uno hacia el otro.

El jugador de la izquierda dispone de las teclas 1 y 2, y el de la derecha de las teclas 9 y Ø.

Si han cambiado los sombreros, las teclas que deberán pulsarse serán 1 (a la izquierda) y Ø (a la derecha).

Si no ha habido cambio, será necesario para ganar pulsar 2 (a la izquierda) y 9 (a la derecha). Entonces, la visualización indicará BRAVO IZQUIERDA o BRAVO DERECHA, o también HA PERDIDO IZQUIERDA o HA PERDIDO DERECHA.

POSIBLES EXTENSIONES: En un aparato como el Spectrum, será posible prever una visualización en color, con animación de los personajes. También podrá complicarse el juego, cambiando varios objetos de forma aleatoria (entonces debería encontrarse la lista completa).

SOMBREROS

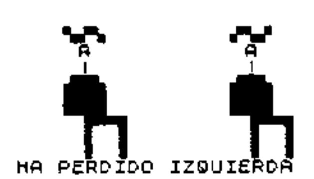
PROGRAMA ZX 81

```
  10 LET B$="▛▚▜"
  20 LET C$="▟▞▙"
  30 LET A$=" ▌  I [redacted] ▌
▌"
  40 LET X=NOT PI
  45 LET T=SGN PI
  50 LET Y=CODE "-"
  60 LET D$=C$
  70 LET C$=B$
  80 LET B$=D$
  90 LET T=ABS (T-SGN PI)
 125 CLS
 130 FOR I=SGN PI TO 7
 140 PRINT AT NOT PI,X;C$;AT NOT
 PI,Y;B$
 150 PRINT AT I,X;A$(INT PI*I-2
TO INT PI*I);AT I,Y;A$(INT PI*I-
2 TO INT PI*I)
 160 NEXT I
 170 LET X=X+SGN PI
 180 LET Y=Y-SGN PI
 182 PAUSE 30
 185 IF X=Y AND RND<.5 THEN GOTO
 CODE "W"
 187 IF INKEY$="" THEN GOTO 125
 190 IF (INKEY$="1" AND T) OR (I
NKEY$="2" AND NOT T) THEN PRINT
"BRAVO IZQUIERDA";W
 200 IF (INKEY$="0" AND T) OR (I
NKEY$="9" AND NOT T) THEN PRINT
"BRAVO DERECHA";W
 210 PRINT "HA PERDIDO ";"IZQUIER
DA" AND (INKEY$="2" OR INKEY$="1
");"DERECHA" AND (INKEY$="9" OR
INKEY$="0")
```



PROGRAMA ZX SPECTRUM

```
  10 LET B$="▛▚▜"
  20 LET C$="▟▞▙"
  30 LET A$=" A  I [redacted] ▌
▌"
  40 LET X=0
  45 LET T=1
  50 LET Y=22
  60 LET D$=C$
  70 LET C$=B$
  80 LET B$=D$
  90 LET T=ABS (T-1)
 125 PAPER 6: CLS
 130 FOR I=1 TO 7
 140 PRINT AT 0,X;C$;AT 0,Y;B$
 150 PRINT INK 2+(I>2 AND I<5);A
T I,X;A$(3*I-2 TO 3*I): PRINT OV
ER 1; INK 5-4*(I>2 AND I<5);AT I
,Y;A$(3*I-2 TO 3*I)
 160 NEXT I
 170 LET X=X+1
 180 LET Y=Y-1
 182 PAUSE 30: IF X=Y THEN BEEP
1,10
 185 IF X=Y AND RND<.5 THEN GO T
O 60
 187 IF INKEY$="" THEN GO TO 125
 190 IF (INKEY$="1" AND T) OR (I
NKEY$="2" AND NOT T) THEN PRINT
FLASH 1;"BRAVO IZQUIERDA": STOP
 200 IF (INKEY$="0" AND T) OR (I
NKEY$="9" AND NOT T) THEN PRINT
FLASH 1;"BRAVO DERECHA": STOP
 210 PRINT "HA PERDIDO IZQUIERDA
" AND (INKEY$="2" OR INKEY$="1")
;"HA PERDIDO DERECHA" AND (INKEY
$="9" OR INKEY$="0")
```



Combate aéreo

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: juego de reflejos (para un jugador).

Dificultad del juego: difícil.

EL JUEGO: En el centro de la pantalla está la mira del aparato, y el avión enemigo se desplaza por delante, de una manera que, al principio, puede parecer incoherente.

El aparato puede desplazarse en todas las direcciones gracias a las flechas de movimiento, y hay que intentar colocar al avión enemigo en el centro del visor.

Hay que disparar sobre él pulsando 0, y se verán converger los proyectiles sobre el avión. Pero continúa desplazándose incluso cuando se le ha disparado, y no está claro que se haya dado en el blanco.

La partida continuará su desarrollo hasta que se le haya tocado (visualización GANA) o haya salido del campo de visión.

EL PROGRAMA: La línea 4Ø dibuja el aparato enemigo, cuyo desplazamiento aleatorio se realiza en las líneas 5Ø y 6Ø.

Las líneas 7Ø y 8Ø permiten pilotar el propio aparato.

La línea 1ØØ dibuja el visor del aparato.

El subprograma que gestiona el disparo va de la línea 21Ø a la línea 28Ø, y la comprobación del final de la partida se efectúa en la línea 27Ø.

La línea 11Ø comprueba si se ha disparado o no.

PARA JUGAR: Desde que se pone en marcha el programa con la orden RUN, debe maniobrarse para intentar encuadrar el aparato enemigo en el visor.

Para esto se dispone de flechas de desplazamiento (teclas 5, 6, 7 y 8), y puede dispararse en cualquier momento pulsando la tecla 0.

El aparato enemigo puede continuar desplazándose de forma aleatoria, aun cuando se le ha disparado. La partida termina con la visualización de GANA si se ha tocado el avión enemigo, o excepcionalmente por el hecho de que el aparato enemigo salga de los límites de la pantalla.

También puede pararse la partida en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: **Será bastante fácil realizar una visualización con un grafismo más cuidado, utilizando color. También podría estabilizarse la posición del avión enemigo durante el tiro (bastará con modificar la línea 16Ø), así como introducir varios aviones, contar los puntos o el tiempo, tener en cuenta la propia reserva de carburante, etc.**

COMBATE AÉREO

PROGRAMA ZX 81

```
  20 LET A=INT PI
  30 LET B=A+A
  32 LET S=SGN PI+SGN PI
  35 LET N=S+S
  40 PRINT AT A,B;"█-█"
  45 PRINT AT A,B;"   "
  50 LET A=A+S*INT (S*RND)-SGN P
I
  60 LET B=B+S*INT (S*RND)-SGN P
I
  70 LET A=A+(INKEY$="6")-(INKEY
$="7")
  80 LET B=B+(INKEY$="8")-(INKEY
$="5")
 100 PRINT AT PI*PI,PI*PI;"▚ ▞";
AT PI*PI+SGN PI,PI*PI;" X";AT PI
*PI+S,PI*PI;"▞ ▚"
 110 IF INKEY$="0" OR N>S+S THEN
 GOSUB CODE "COS "
 120 GOTO CODE "C"
 210 PLOT S+N,N
 220 PLOT CODE "F"-N,N
 230 UNPLOT N,N-S
 240 UNPLOT CODE "H"-N,N-S
 250 LET N=N+SGN PI
 260 IF N<CODE "+" THEN RETURN
 270 IF A=11 AND B=10 THEN PRINT
 "GAGNE";W
 280 GOTO CODE "7"
```

PROGRAMA ZX SPECTRUM

```
  20 LET A=3
  30 LET B=A+A
  32 LET S=2
  35 LET N=S+S
  40 PRINT INK 2;AT A,B;"█-█"
  45 LET U=A: LET V=B
  50 LET A=A+S*INT (S*RND)-1
  60 LET B=B+S*INT (S*RND)-1
  70 LET A=A+S*((INKEY$="6")-(IN
KEY$="7"))
  80 LET B=B+S*((INKEY$="8")-(IN
KEY$="5"))
  90 PRINT AT U,V;"   "
 100 PRINT INK 1;AT 9,9;"▚ ▞";AT
 10,10;"X";AT 11,9;"▞ ▚"
 110 IF INKEY$="0" OR N>S+S THEN
 GO SUB 200
 120 GO TO 40
 210 PLOT 4*(S+N-3),4*N
 220 PLOT  172-4*N,4*N
 230 PLOT OVER 1,4*(N-3),4*(N-S)
 240 PLOT OVER 1,180-4*N,4*(N-S)
 250 LET N=N+1
 260 IF N<21 THEN RETURN
 270 IF (A=11 OR A=10 OR A=12) A
ND (B=10 OR B=9 OR B=11) THEN PR
INT FLASH 1; PAPER 4;"GANADO..."
: BEEP .3,10: BEEP .3,18: BEEP .
3,7: STOP
 280 GO TO 35
```



Dibujo de dados

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: difícil.
Categoría del juego: tirada de dados.
Dificultad del juego: fácil.

EL JUEGO: El programa propuesto es interesante sobre todo por su aspecto gráfico, ya que permite representar en la pantalla, de una manera muy acertada, de 1 a 4 dados, utilizando una técnica de dibujo algo particular.

Cada respuesta a la pregunta NÚM. DE DADOS provoca inmediatamente una nueva tirada del número de dados indicado.

EL PROGRAMA: La línea 3Ø permite controlar si se ha pulsado o no una de las teclas del 1 al 4.

Las líneas 5Ø y 6Ø inicializan las dos variables del dibujo de los dados.

El bucle I (7Ø-14Ø) y las líneas 8Ø y 1ØØ dibujan el cuadrado del dado.

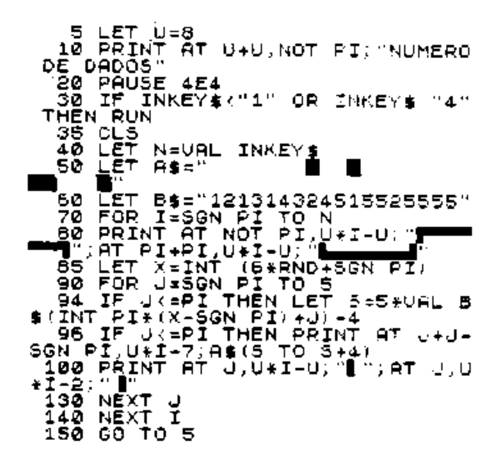
La línea 85 realiza la tirada, y el bucle J (9Ø-13Ø) dibuja los puntos en relación con la tirada efectuada (llamando a las variables A$ y B$).

PARA JUGAR: Se pone en marcha el programa con la orden RUN, y a la pregunta NÚM. DE DADOS, se responde pulsando una tecla de 1 a 4. Entonces, los dados que se han pedido se representan, y vuelve a realizarse la pregunta NÚM. DE DADOS. Bastará con pulsar BREAK para salir del programa.

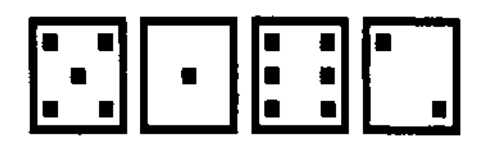
POSIBLES EXTENSIONES: Podrá conservarse la parte del dibujo que se ha utilizado en este programa para realizar todo tipo de juegos de dados, con una representación gráfica de calidad conveniente y fácil de modificar, si fuera necesario.

DIBUJO DE DADOS

PROGRAMA ZX 81

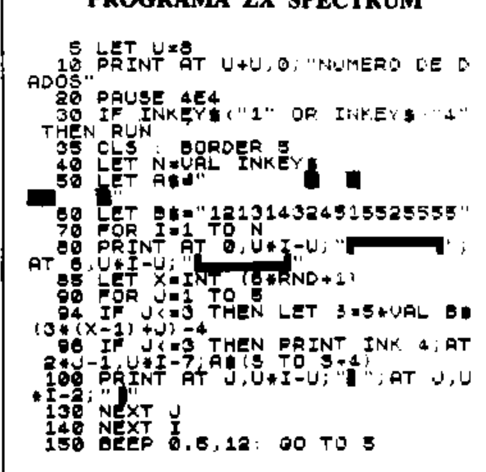


```
   5 LET U=8
  10 PRINT AT U+U,NOT PI;"NUMERO
 DE DADOS"
  20 PAUSE 4E4
  30 IF INKEY$<"1" OR INKEY$>"4"
 THEN RUN
  35 CLS
  40 LET N=VAL INKEY$
  50 LET A$="          ■  ■
■    ■"
  60 LET B$="121314324515525555"
  70 FOR I=SGN PI TO N
  80 PRINT AT NOT PI,U*I-U;"▛▀▀▀▀
▀▀▜";AT PI+PI,U*I-U;"▙▄▄▄▄▄▄▟"
  85 LET X=INT (6*RND+SGN PI)
  90 FOR J=SGN PI TO 5
  94 IF J<=PI THEN LET S=5*VAL B
$(INT PI*(X-SGN PI)+J)-4
  96 IF J<=PI THEN PRINT AT U+J-
SGN PI,U*I-7;A$(S TO S+4)
 100 PRINT AT J,U*I-U;"▌";AT J,U
*I-2;"▐"
 130 NEXT J
 140 NEXT I
 150 GO TO 5
```

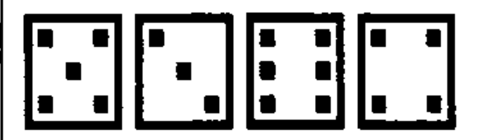


NUMERO DE DADOS

PROGRAMA ZX SPECTRUM



```
   5 LET U=8
  10 PRINT AT U+U,0;"NUMERO DE D
ADOS"
  20 PAUSE 4E4
  30 IF INKEY$<"1" OR INKEY$>"4"
 THEN RUN
  35 CLS : BORDER 5
  40 LET N=VAL INKEY$
  50 LET A$="          ■  ■
■    ■"
  60 LET B$="121314324515525555"
  70 FOR I=1 TO N
  80 PRINT AT 0,U*I-U;"▛▀▀▀▀▀▀▜";
AT 6,U*I-U;"▙▄▄▄▄▄▄▟"
  85 LET X=INT (6*RND+1)
  90 FOR J=1 TO 5
  94 IF J<=3 THEN LET S=5*VAL B$
(3*(X-1)+J)-4
  96 IF J<=3 THEN PRINT INK 4;AT
 2*J-1,U*I-7;A$(S TO S+4)
 100 PRINT AT J,U*I-U;"▌";AT J,U
*I-2;"▐"
 130 NEXT J
 140 NEXT I
 150 BEEP 0.6,12: GO TO 5
```



NUMERO DE DADOS

El juego del misil

REFERENCIAS:

Modalidad: FAST.

Dificultad de programación: mediana.

Categoría del juego: contra el ZX.

Dificultad del juego: mediana.

EL JUEGO:

El ZX está alerta: el jugador está a los mandos de un avión encargado de una misión de espionaje sobre su territorio; el objetivo es sobrevolar una base enemiga (representada por una H invertida) y regresar sano y salvo a la base propia; pero en cuanto el ZX haya localizado el avión enviará en su persecución un misil muy perfeccionado que no parará hasta destruirlo.

El avión puede desplazarse en todas las direciones gracias a las teclas 1 a 8; ¿logrará el jugador completar su misión?

EL PROGRAMA:

La línea 3Ø dibuja la base enemiga que hay que sobrevolar (bajo la forma de una H invertida).

La línea 4Ø dibuja la base de salida.

Las líneas 55 y 65 escogen el instante de la salida del misil.

La línea 7Ø dibuja el avión.

Las líneas 1ØØ y 12Ø definen su movimiento siguiendo 8 direcciones, y utilizando los subprogramas de las líneas 1ØØØ, 2ØØØ, 3ØØØ y 4ØØØ.

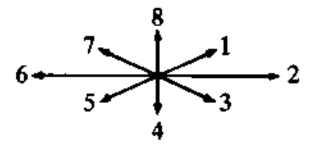
La línea 125 garantiza la salida del misil, que se dibuja por la línea 13Ø y cuyo desplazamiento es gestionado por las líneas 135 y 14Ø.

Las líneas 145 y 15Ø controlan el final de la partida.

PARA JUGAR:

Poner en marcha el programa con RUN.

Se visualiza el terreno de juego, y el jugador puede desplazar el aparato utilizando las teclas de la 1 a la 8 en las direcciones siguientes:



Debe rodearse el territorio enemigo (si el jugador o el misil tocan este territorio, se empezará una nueva partida).

Si el misil destruye el avión, se obtendrá el mensaje TOCADO y la partida terminará.

Pulsar BREAK para acabar la partida.

POSIBLES EXTENSIONES: Sin duda será necesario añadir diversos controles al programa (el avión ha sobrevolado la base enemiga, ha regresado a su propia base...).

También pueden preverse diversos niveles de juego, por ejemplo, variando la velocidad del misil, prever la utilización de señuelos por el avión, la explosión del misil al cabo de un cierto tiempo, etc.

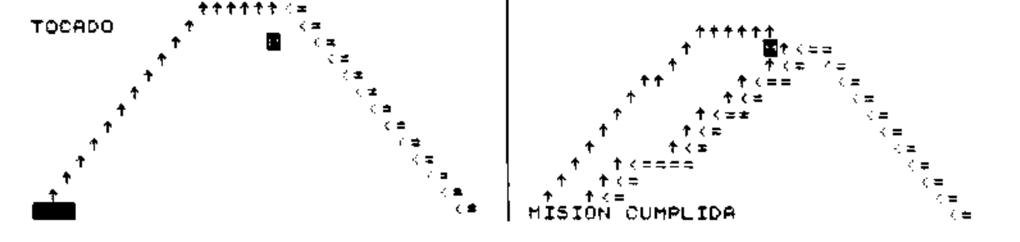
EL JUEGO DEL MISIL

PROGRAMA ZX 81

```
   5 CLS
  10 LET A=SGN PI+SGN PI
  20 LET B=A+INT PI
  30 PRINT AT B,CODE ")";"■"
  40 PRINT AT CODE "?",NOT PI,"■
■"
  47 LET U=CODE "£"
  50 LET X=CODE ":"
  55 LET S=INT (RND*U)
  60 LET Y=SGN PI
  65 LET S=S-SGN PI
  67 LET V=CODE "3"
  70 PRINT AT X,Y;"+"
  90 PAUSE 4E4
 100 LET F=A*(VAL INKEY$(B)-SGN
PI
 120 GO SUB 1000*(VAL INKEY$+A*(
F-SGN PI))
 125 IF S THEN GO TO CODE "INKEY
$"
 130 PRINT AT U,V;"<="
 135 LET U=U-SGN (U-X)
 140 LET V=V-SGN (V-Y)
 145 IF U=X AND V=Y THEN PRINT "
TOCADO";W
 150 IF U=B AND V=CODE ")" THEN
RUN
 200 GO TO 70
1000 LET X=X-F
2000 LET Y=Y+F
2010 RETURN
3000 GO SUB 2000
4000 LET X=X+F
4010 RETURN
```

PROGRAMA ZX SPECTRUM

```
   5 PAPER 7: CLS
  10 LET A=2
  15 LET M=0
  20 LET B=6
  30 PRINT AT B,17;"■"
  40 PRINT INK 4;AT 15,0;"■■■"
  47 LET U=15
  50 LET X=14
  55 LET S=INT (RND*U)
  60 LET Y=1
  65 LET S=S-1
  67 LET V=30
  70 PRINT AT X,Y;"↑"
  90 PAUSE 4E4
 100 LET F=A*(VAL INKEY$(B)-1
 120 GO SUB 1000*(VAL INKEY$+A*(
F-1))
 125 IF S THEN GO TO 65
 130 PRINT PAPER 5;AT U,V;"<="
 135 LET U=U-SGN (U-X)
 140 LET V=V-SGN (V-Y)
 145 IF U=X AND V=Y THEN PRINT F
LASH 1;"TOCADO": BEEP 1,0: STOP
 150 IF U=B AND V=17 THEN RUN
 160 IF X<=B AND Y>=17 THEN LET
M=1
 170 IF M AND X=15 AND Y<=3 THEN
 PRINT INK 0; FLASH 1;"MISION CU
MPLIDA": BEEP 1,10: STOP
 200 GO TO 70
1000 LET X=X-F
2000 LET Y=Y+F
2010 RETURN
3000 GO SUB 2000
4000 LET X=X+F
4010 RETURN
```



El mono

REFERENCIAS:

Modalidad: SLOW o FAST.

Dificultad de programación: mediana.

Categoría del juego: dos jugadores.

Dificultad del juego: difícil.

EL JUEGO:

Se inspira en una antigua idea del mono que golpeando al azar sobre una máquina de escribir tiene posibilidades de acabar escribiendo un texto con sentido.

El ZX realiza aquí el papel del mono y va a visualizar grupos de 3 a 6 letras durante un tiempo relativamente corto. Si uno de los jugadores cree que logrará componer una palabra con las letras propuestas, tiene que pulsar rápidamente la tecla de juego (1 o Ø).

Entonces el adversario aceptará o rechazará la palabra y responderá CORRECTO o FALSO. Cada palabra correcta tiene 1 punto, cada palabra falsa resta 1 punto.

EL PROGRAMA:

La línea 3Ø contiene una lista V$ de vocales corrientes y la línea 4Ø una lista C$ de consonantes corrientes.

Será posible modificar estas dos listas con el fin de aumentar, por ejemplo, la frecuencia de aparición de algunas letras (de esta forma podrán ponerse dos E en V$ o dos R en C$, etc.).

La línea 5Ø define la longitud de la palabra (en principio de 3 a 6 letras, pero también puede modificarse la longitud).

El bucle I (8Ø-13Ø) permite la confección de una "palabra" tomando una letra del alfabeto completo en la lista C$ (línea 1ØØ) o en la lista V$ (línea 11Ø).

La línea 15Ø contemporiza la aparición del resultado.

La línea 16Ø gestiona el final de la partida con la visualización de la puntuación.

La línea 17Ø comprueba si uno de los dos jugadores ha reaccionado.

Por último, las líneas 22Ø y 23Ø contabilizan los puntos de cada jugador.

PARA JUGAR:

Se pone en marcha el programa pulsando RUN.

La primera serie de letras se visualiza entonces en la pantalla. El jugador de la izquierda deberá pulsar 1 si ve que puede componer una palabra; el de la derecha pulsará Ø.

Si no se ha pulsado nada se visualizará otra serie de letras poco rato después.

Si uno de los jugadores ha pulsado Ø o 1, se visualiza la pregunta C o F. Se responde C si la palabra es correcta, y F si la palabra no es correcta, seguido de NEWLINE. Entonces, el ZX calcula los puntos y la siguiente serie de letras se visualiza.

Para salir del programa deberá pulsarse P en lugar de Ø o 1: se visualizará la puntuación final.

POSIBLES EXTENSIONES: Podrá modificarse la longitud de las palabras, la frecuencia de aparición de las letras, el descuento de puntos, la regla del juego, etc.

EL MONO

PROGRAMA ZX 81

```
  10 LET A=NOT PI
  20 LET B=A
  30 LET V$="AEIOU"
  40 LET C$="BCDGLMNPRSTV"
  50 LET X=INT (4*RND+INT PI)
  60 LET M$=""
  70 PRINT AT 5,NOT PI;"          "
  80 FOR I=SGN PI TO X
  90 LET E$=CHR$ (INT (26*RND+38
))
 100 IF RND<.4 THEN LET E$=C$(IN
T (LEN C$*RND+SGN PI))
 110 IF RND<.5 THEN LET E$=V$(IN
T (LEN V$*RND+SGN PI))
 120 LET M$=M$+E$
 130 NEXT I
 140 PRINT AT 5,NOT PI;M$
 150 PAUSE 60
 160 IF INKEY$="P" THEN PRINT AT
 7,NOT PI;"MARCADOR DE A :";A;";
 DE B :";B;W
 170 IF INKEY$<>"1" AND INKEY$<>
"0" THEN GOTO 50
 180 LET T$=INKEY$
 190 PRINT AT 7,NOT PI;"C O F"
 200 INPUT R$
 210 PRINT AT 7,NOT PI;"          "
 220 LET A=A+(R$="C" AND T$="0")
-(R$="F" AND T$="0")
 230 LET B=B+(R$="C" AND T$="1")
-(R$="F" AND T$="1")
 240 GOTO 50
```

```
GCUSA

MARCADOR DE A :-2; DE B :3
```

PROGRAMA ZX SPECTRUM

```
  10 LET A=0
  20 LET B=A
  30 LET V$="AEIOU"
  40 LET C$="BCDGLMNPRSTV"
  50 LET X=INT (4*RND+3)
  60 LET M$=""
  70 PRINT AT 5,0;"          "
  80 FOR I=1 TO X
  90 LET E$=CHR$ INT (26*RND+65)
 100 IF RND<.4 THEN LET E$=C$(IN
T (LEN C$*RND+1))
 110 IF RND<.5 THEN LET E$=V$(IN
T (LEN V$*RND+1))
 120 LET M$=M$+E$
 130 NEXT I
 140 PRINT PAPER 5; FLASH 1;AT 5
,0;M$
 150 PAUSE 60
 160 IF INKEY$="P" THEN PRINT FL
ASH 1;AT 7,0;"MARCADOR DE A :";A
;"  MARCADOR DE B :";B: BEEP 1,1
0: STOP
 170 IF INKEY$<>"1" AND INKEY$<>
"0" THEN GO TO 50
 180 LET T$=INKEY$
 190 PRINT AT 7,0; FLASH 1;"(C)"
; FLASH 0;"ORRECTO O "; FLASH 1;
"(F)"; FLASH 0;"ALSO ?"
 200 INPUT R$
 210 PRINT AT 7,0;"

"
 220 LET A=A+(R$="C" AND T$="0")
-(R$="F" AND T$="0")
 230 LET B=B+(R$="C" AND T$="1")
-(R$="F" AND T$="1")
 240 GO TO 50
```

```
EONK

MARCADOR DE A :4  MARCADOR DE B
:-2
```

```
EONK

MARCADOR DE A :4  MARCADOR DE B
:-2
```

El juego del 15

REFERENCIAS:

Modalidad: SLOW.

Dificultad de programación: difícil.

Categoría del juego: de reflexión (dos jugadores).

Dificultad del juego: mediana.

EL JUEGO:

Es un juego bastante antiguo para dos jugadores, que hace tiempo se practicaba en las ferias. Se dispone de un tapete de juego de 9 casillas, y por turnos cada persona apuesta una ficha de su color en una casilla libre.

El primero de los dos jugadores que con las tres fichas exactas logre hacer un total de 15 ha ganado.

EL PROGRAMA:

El bucle I (40-65) dibuja el campo de juego, y la línea 66 se ocupa de la gestión del turno de los jugadores (número 1 o 2 para cada jugador y la ficha 0 o X correspondiente).

La línea 90 espera la jugada (podría utilizarse el programa en modalidad FAST, reemplazando esta línea por una pausa).

El bucle (100-120) verifica la validez de la jugada.

La línea 122 comprueba el fin del juego en caso de empate, y la línea 130 rechaza las jugadas que no son correctas (con regreso a la situación inicial, línea 66).

Las líneas 140 y 143 graban las respectivas jugadas de cada jugador en las variables alfanuméricas O$ y X$.

La línea 145 se ocupa de la visualización de la jugada, y la línea 150 la graba en una variable R$.

La parte más interesante del programa, que podría utilizarse para definir una estrategia de máquina, va de la línea 157 a la línea 180: se utilizan 3 bucles intercalados para verificar si uno de los jugadores ha obtenido con 3 fichas un total de 15.

El cambio de jugador se efectúa en la línea 190.

PARA JUGAR:

Poner en marcha el juego pulsando RUN.

Se visualiza el terreno de juego y después la pregunta JR 1 : X, y el jugador que ha escogido las x tendrá que dar el número de la casilla donde quiere jugar; si está libre, su jugada será grabada.

A continuación se obtiene la visualización JR 2 : 0, y el segundo jugador procede de la misma forma.

Si uno de los jugadores gana (total de 15 sobre 3 jugadas), obtiene la visualización BRAVO y la partida se para.

En caso de empate, la máquina visualizará HA TERMINADO.

Se empezará otra partida pulsando RUN.

POSIBLES EXTENSIONES: Sin ninguna duda, la más interesante será definir una estrategia de máquina (tratando de evitar que la máquina juegue siempre el 5 cuando debe empezar).

Naturalmente, también puede mejorarse el grafismo e introducir efectos sonoros, para los distintos mensajes.

EL JUEGO DEL 15

PROGRAMA ZX 81

```
  10 LET F=NOT PI
  20 LET R$="123456789"
  30 LET O$=""
  35 LET X$=""
  40 FOR I=F TO CODE "?" STEP IN
T PI
  50 PRINT AT INT PI,I;"▄▄▄"
  60 PRINT AT INT PI+SGN PI,I+SG
N PI;(I+INT PI)/INT PI
  65 NEXT I
  66 PRINT AT CODE "£",NOT PI;"J
R ";"O" AND F;"X" AND NOT F
  90 IF INKEY$<"1" OR INKEY$>"9"
 THEN GOTO 90
 100 FOR I=SGN PI TO LEN R$
 115 IF INKEY$=R$(I) THEN GOSUB
140
 120 NEXT I
 122 IF NOT VAL R$ THEN PRINT "H
A TERMINADO";U
 130 GOTO CODE "PI"
 140 LET O$=O$+(INKEY$ AND F)
 143 LET X$=X$+(INKEY$ AND NOT F
)
 145 PRINT AT INT PI-SGN PI,INT
PI*I-SGN PI-SGN PI;"X" AND NOT F
;"O" AND F
 150 LET R$(I)="0"
 157 LET T$=(O$ AND F)+(X$ AND N
OT F)
 158 IF LEN T$<INT PI THEN GOTO
CODE "█"
 160 FOR I=SGN PI TO INT PI
 162 FOR J=I+SGN PI TO LEN T$
 164 FOR K=J+SGN PI TO LEN T$
 166 IF VAL T$(I)+VAL T$(J)+VAL
T$(K)=CODE "?" THEN PRINT "BRAVO
";U
 170 NEXT K
 175 NEXT J
 180 NEXT I
 190 LET F=ABS (F-SGN PI)
 200 RETURN
```



JR X

PROGRAMA ZX SPECTRUM

```
  10 LET F=0
  20 LET R$="123456789"
  30 LET O$=""
  35 LET X$=""
  40 FOR I=F TO 24 STEP 3
  50 PRINT INK 5;AT 1,I;"▀▀▀";AT
 2,I;"▌ ▐";AT 3,I;"▄▄▄"
  60 PRINT AT 4,I+1;(I+3)/3
  65 NEXT I
  66 PRINT AT 12,0;"JUGADOR ";F+
1;" :O" AND F;" :X" AND NOT F
  90 IF INKEY$<"1" OR INKEY$>"9"
 THEN GO TO 90
 100 FOR I=1 TO LEN R$
 115 IF INKEY$=R$(I) THEN GO SUB
 140
 120 NEXT I
 122 IF NOT VAL R$ THEN PRINT FL
ASH 1; INK 2;AT 18,0;"HA TERMINA
DO": STOP
 130 GO TO 66
 140 LET O$=O$+(INKEY$ AND F)
 143 LET X$=X$+(INKEY$ AND NOT F
)
 145 PRINT AT 2,3*I-2;"X" AND NO
T F;"O" AND F: BEEP .3,10+10*F
 150 LET R$(I)="0"
 157 LET T$=(O$ AND F)+(X$ AND N
OT F)
 158 IF LEN T$<3 THEN GO TO 190
 160 FOR I=1 TO 3
 162 FOR J=I+1 TO LEN T$
 164 FOR K=J+1 TO LEN T$
 166 IF VAL T$(I)+VAL T$(J)+VAL
T$(K)=15 THEN PRINT FLASH 1; INK
 2;AT 18,0;"BRAVO , JUGADOR ";"O
" AND F;"X" AND NOT F: STOP
 170 NEXT K
 175 NEXT J
 180 NEXT I
 190 LET F=ABS (F-1)
 200 RETURN
```



JUGADOR 1 :X

BRAVO , JUGADOR X

Destreza

REFERENCIAS:

Modalidad: FAST.
Dificultad de programación: difícil.
Categoría del juego: reflejos (para un jugador).
Dificultad del juego: difícil.

EL JUEGO:

Van a aparecer cifras al azar entre 0 y 9 en diferentes sitios de la pantalla.

Quedan visualizadas durante un tiempo muy corto.

El jugador debe lograr pulsar la tecla numérica correspondiente en el mismo instante en que la cifra se visualiza.

La cifra pulsada por el jugador se visualizará en tamaño grande, a la izquierda de la pantalla, y cada respuesta exacta proporciona 10 puntos.

La duración de la partida es aleatoria, y la puntuación se visualizará al final de la partida.

EL PROGRAMA:

Las líneas 4Ø y 5Ø contienen, en dos variables alfanuméricas C$ y B$, las referencias que serán utilizadas para dibujar las cifras en tamaño grande.

La situación de la cifra que saca el ZX se escoge en las líneas 6Ø y 7Ø, y la cifra se saca en la línea 8Ø.

La línea 9Ø realiza la impresión, y la línea 1ØØ determina la duración de la visualización.

La línea 12Ø define la duración del juego, y el final de la partida es tratado por la línea 13Ø.

El bucle I (21Ø-24Ø) realiza el dibujo de la cifra pulsada por el jugador en tamaño grande.

El control de la exactitud del resultado es efectuado por la línea 25Ø.

PARA JUGAR:

El programa se pondrá en marcha por RUN; después de una corta pausa, aparece la primera cifra hacia el centro de la pantalla. El jugador tendrá que reaccionar bastante rápido durante el tiempo que dure la visualización y volver a pulsar la cifra.

La cifra que se ha pulsado se dibujará entonces en tamaño grande en la parte superior y a la izquierda de la pantalla.

Si el resultado es exacto, el jugador marcará 10 puntos.

El tiempo de desarrollo de la partida estará en función de las cifras que aparezcan en la visualización.

La puntuación se visualizará al final de la partida.

Será posible parar en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: **Puede considerarse modificar el juego para hacer participar a varios jugadores, cambiando eventualmente la visualización y codificando las teclas del teclado, puestas a disposición de cada uno de los jugadores. También podrán obtenerse puntos con cada resultado falso, modificar las reglas del cálculo de la puntuación y la duración del juego, etc.**

DESTREZA

PROGRAMA ZX 81

```
   5 LET M=10
  10 LET T=M*M
  20 PAUSE T
  30 LET S=NOT PI
  40 LET C$="███    █ ▌"
  50 LET B$="1444133333131211313
144133121311214113333141411 4131"
  60 LET X=INT (M*RND)
  70 LET Y=INT (M*RND+M)
  80 LET N=INT (M*RND)
  90 PRINT AT X,Y;N
 100 PAUSE INT (40*RND+M+M+M)
 110 IF INKEY$<>"" THEN GO TO CO
DE "COS "
 120 LET T=T-N
 125 PRINT AT X,Y;" "
 130 IF T>=NOT PI THEN GO TO 60
 140 CLS
 150 PRINT "MARCADOR ";S;"/100"
 160 STOP
 200 LET U=VAL INKEY$
 210 FOR I=SGN PI TO 5
 220 LET A=2*VAL B$(5*U+I)-SGN P
I
 230 PRINT AT I,NOT PI;C$(A TO A
+SGN PI)
 240 NEXT I
 250 IF U=N THEN LET S=S+M
 260 GO TO 120
```

3

4

PROGRAMA ZX SPECTRUM

```
   5 LET M=10
  10 LET T=M*M
  20 PAUSE T: BORDER 6: CLS
  30 LET S=0
  40 LET C$="███    █ ▌"
  50 LET B$="1444133333131211313
1441331213112141133331414114131"
  60 LET X=INT (M*RND)
  70 LET Y=INT (M*RND+M)
  80 LET N=INT (M*RND)
  90 PRINT AT X,Y;N
 100 PAUSE INT (40*RND+M+M+M)
 110 IF INKEY$<>"" THEN GO TO 20
0
 120 LET T=T-N
 125 PRINT AT X,Y;" "
 130 IF T>=0 THEN GO TO 60
 140 CLS : BEEP .3,10: BEEP .3,1
5: BEEP .3,5
 150 PRINT "MARCADOR ";S;"/100"
 160 PAUSE 4E4: RUN
 200 LET U=VAL INKEY$
 210 FOR I=1 TO 5
 220 LET A=2*VAL B$(5*U+I)-1
 230 PRINT INK INT (4*RND+1);AT
I,0;C$(A TO A+1)
 240 NEXT I
 250 IF U=N THEN LET S=S+M: BEEP
 .1,15
 260 GO TO 120
```

5

6

Submarino

REFERENCIAS:

Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: reflejos (un solo jugador).

Dificultad del juego: fácil.

EL JUEGO:

Se maniobra en superficie una nave cazadora de submarinos, y éstos pasan regularmente por debajo de la unidad a profundidades diferentes.

Pueden lanzarse en cualquier momento granadas submarinas, e incluso se puede intentar tocar el mismo submarino varias veces.

La puntuación se visualizará permanentemente a la izquierda de la pantalla (relación del número de disparos al blanco sobre el número de submarinos aparecidos).

EL PROGRAMA:

La línea 2Ø permite al ZX escoger la profundidad del submarino.

La línea 5Ø dibuja la embarcación.

La línea 6Ø muestra el tanteo y dibuja la superficie del agua.

La línea 8Ø dibuja el submarino.

El disparo se efectúa por un programa específico, situado en las líneas 2ØØ a 225.

El desplazamiento (muy rápido) de la nave se realiza por la línea 11Ø.

La comprobación que permite saber si se ha tocado o no se efectúa en la línea 85.

PARA JUGAR:

Pulsar RUN para empezar la partida.

La nave se desplazará gracias a las flechas derecha e izquierda (5 y 8).

La tecla 0 autoriza el disparo.

Como la nave se desplaza tres veces más rápida que los submarinos, será posible tocar varias veces la misma embarcación.

Se pulsará BREAK para terminar la partida.

POSIBLES EXTENSIONES: Será elemental mejorar la calidad gráfica del juego, utilizar color y sonidos, hacer aparecer varios submarinos que se desplacen a distintas velocidades, y hasta llegado el caso permitir a los submarinos que también disparen, etc.

SUBMARINO

PROGRAMA ZX 81

```
  6 LET S=NOT PI
 12 LET C=CODE "?"
 14 LET P=5
 15 PRINT AT S,NOT PI;"      "
 17 LET T=CODE "."
 20 LET S=INT (6*RND+PI)
 25 LET P=P+.01
 35 LET X=SGN PI
 50 PRINT AT SGN PI,C;"   ▄█▄/
"
 60 PRINT P;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
 80 PRINT AT S,T;" ▄█▄ "
 82 IF X=SGN PI THEN LET Y=C+X+
X
 85 IF S=X AND T+SGN PI=Y THEN
LET P=P+SGN PI
 90 LET T=T-SGN PI
100 IF T<NOT PI THEN GOTO CODE
"?"
110 LET C=C+INT PI*(INKEY$="8"
AND C<CODE "+")-INT PI*(INKEY$="
5" AND C>NOT PI)
120 IF INKEY$<>"0" AND X=SGN PI
 THEN GOTO CODE "M"
200 PRINT AT X,Y;" "
210 LET X=X+SGN PI
215 IF X>S THEN GOTO CODE "7"
220 PRINT AT X,Y;"O"
225 GOTO CODE "M"
```

```
2.03▀▀▀▀▀▀▀▀▄█▄/▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀


                          ▄█▄
```

PROGRAMA ZX SPECTRUM

```
  6 LET S=0
 12 LET C=15
 14 LET P=5
 15 PRINT AT 5,0;"      "
 17 LET T=27
 20 LET S=INT (6*RND+4)
 25 LET P=P+0.01
 35 LET X=1
 50 PRINT INK 2;AT 1,C;"   ▄█▄
/
 60 PRINT PAPER 5; FLASH 1;AT 1
6,1;"MARCADOR : ";P; FLASH 0; IN
K 1;AT 2,0;"████████████████████
"
 80 PRINT INK 6;AT S,T;" ▄█▄ "
 82 IF X=1 THEN LET Y=C+X+X
 85 IF S=X AND (T+2=Y OR T+1=Y)
 THEN LET P=P+1: PRINT INK 3; FL
ASH 1;AT S,T;"***": BEEP 1,-15
 90 LET T=T-1: IF P=99 THEN PRI
NT INVERSE 1; INK 4;AT 12,0;"HA
TERMINADO": BEEP .3,10: BEEP .3,
10: BEEP .3,5
100 IF T<0 THEN GO TO 15
110 LET C=C+3*((INKEY$="8" AND
C<21)-(INKEY$="5" AND C>0))
120 IF INKEY$<>"0" AND X=1 THEN
 GO TO 50
200 PRINT AT X,Y;" "
210 LET X=X+1
215 IF X>S THEN GO TO 35
220 PRINT AT X,Y;"O"
225 GO TO 50
```

```
████████████████████▄█▄/████████


        ▄█▄



  MARCADOR : 20.32
```

Zombis

REFERENCIAS: Modalidad: FAST o SLOW.
Dificultad de programación: bastante difícil.
Categoría del juego: solitario.
Dificultad del juego: bastante fácil.

EL JUEGO: Los zombis son unos seres maléficos que intentarán destruir al jugador, pero éste intentará utilizarlos para eliminar de su espacio vital unos agujeros negros que lo oscurecen. En la pantalla se traza un cuadrado que define el espacio vital.

En este cuadrado aparecerán de forma aleatoria los ZOMBIS (que son Z invertidas), los agujeros negros (cuadrados negros) y el jugador (un Ø).

Los ZOMBIS convergen hacia el jugador intentando capturarlo, pero éste se desplaza más rápido que ellos e intentará hacerles comer todos los agujeros negros, sin dejarse capturar. La partida se termina por la desaparición del jugador o por la eliminación de todos los agujeros negros.

EL PROGRAMA: La posición de los ZOMBIS está definida por una tabla Z en la línea 1Ø.

Las líneas 4Ø y 8Ø (bucle I) dibujan el cuadrado, los ZOMBIS y los agujeros negros.

La línea 92 define la posición inicial del jugador.

Un segundo bucle I (95-1ØØ) realiza el desplazamiento de las Z.

Las líneas 12Ø y 13Ø controlan el desplazamiento del jugador.

PARA JUGAR: Después de poner en marcha la ejecución con RUN, el jugador se desplazará utilizando las 4 flechas y teniendo cuidado de no salir de los límites del terreno.

Tendrá que maniobrar de manera que los ZOMBIS vayan sobre los agujeros negros, sin dejarse capturar él mismo.

Para terminar la partida, bastará con pulsar BREAK.

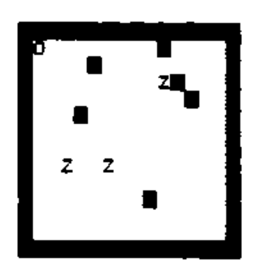
POSIBLES EXTENSIONES: Sería interesante aumentar el tamaño del terreno de juego, aumentar el número de ZOMBIS o de agujeros negros, mejorar los controles (prácticamente inexistentes), situar al jugador de forma más aleatoria, eventualmente modificar la regla (por ejemplo, hacer desaparecer los ZOMBIS en los agujeros negros).

También podrá mejorarse la calidad del dibujo e introducir color.

ZOMBIS

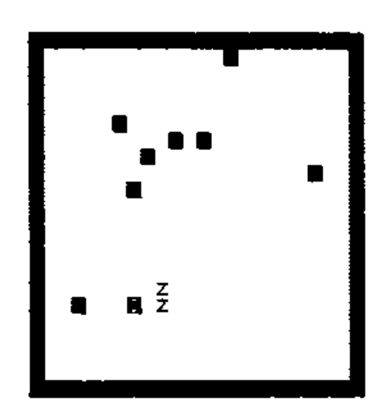
PROGRAMA ZX 81

```
  10 DIM Z(INT PI+INT PI)
  20 LET L=NOT PI
  40 FOR I=L TO CODE "$"
  50 PRINT AT L,I+SGN PI;"■"
  60 PRINT AT I,L;"■";AT I,CODE
"?";"■"
  65 IF RND<.4 THEN PRINT AT INT
 (RND*CODE "$"),I;"■"
  67 LET S=RND*VAL "10"
  69 LET Z(INT ((I+SGN PI+SGN PI
)/INT PI)+SGN PI)=INT PI+INT S
  75 PRINT AT CODE "$",I+SGN PI;
"■"
  80 NEXT I
  85 LET L=SGN PI+SGN PI
  91 LET K=S
  92 PRINT AT S,K;"O"
  95 FOR I=SGN PI TO INT PI
  96 PRINT AT Z(I),Z(I+INT PI);"
 "
  97 LET Z(I)=Z(I)-SGN (Z(I)-S)
  98 LET Z(I+INT PI)=Z(I+INT PI)
-SGN (Z(I+INT PI)-K)
  99 PRINT AT Z(I),Z(I+INT PI);"
Z"
 100 NEXT I
 110 PAUSE I*I
 115 PRINT AT S,K;" "
 120 LET K=K+L*((INKEY$="8")-(IN
KEY$="5"))
 130 LET S=S+L*((INKEY$="6")-(IN
KEY$="7"))
 190 GO TO 92
```



PROGRAMA ZX SPECTRUM

```
  10 DIM Z(8)
  20 LET L=0
  40 FOR I=L TO 21
  50 PRINT INK 3;AT L,I+1;"■"
  60 PRINT INK 3;AT I,L;"■";AT I
,23;"■"
  65 IF RND<.4 THEN PRINT AT INT
 (RND*20)+1,I;"■"
  67 LET S=INT (RND*14+1)
  69 LET Z(INT (I/3)+1)=5+S
  75 PRINT INK 3;AT 21,I+1;"■"
  80 NEXT I
  85 LET L=2
  91 LET K=INT (RND*18+1)
  95 FOR I=1 TO 4
  96 PRINT AT Z(I),Z(I+4);" "
  97 LET Z(I)=Z(I)-SGN (Z(I)-S)
  98 LET Z(I+4)=Z(I+4)-SGN (Z(I+
4)-K)
  99 PRINT INK 2; FLASH 1;AT Z(I
),Z(I+4);"Z": IF Z(I)=S AND Z(I+
4)=K THEN BEEP .3,18: BEEP .3,18
: BEEP .3,18: PRINT FLASH 1;"HA
PERDIDO": STOP
 100 NEXT I
 105 PRINT PAPER 5; INVERSE 1;AT
 S,K;"O"
 110 PAUSE I↑I
 115 PRINT AT S,K;" "
 120 LET K=K+L*((INKEY$="8")-(IN
KEY$="5")): IF K<=0 OR K>=23 THE
N PRINT "HA PERDIDO": BEEP 3,-10
: STOP
 130 LET S=S+L*((INKEY$="6")-(IN
KEY$="7")): IF S<=0 OR S>=21 THE
N PRINT "HA PERDIDO": BEEP 3,-10
: STOP
 190 GO TO 92
```



Diago

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: fácil.
Categoría del juego: reflejos (un jugador).
Dificultad del juego: difícil.

EL JUEGO: Se pone en marcha el programa por RUN y se obtiene una primera visualización que informa sobre la marca a realizar (por ejemplo, 5 puntos negros o 17 puntos negros); después se visualiza el tiempo y se descuenta a partir de 1Ø segundos.

El juego empieza cuando se llega a 0 segundos. Entonces aparecen unos puntos negros, abajo y a la izquierda de la pantalla, formando una línea en diagonal, y habrá que pulsar la tecla Ø para parar la visualización de los puntos tan pronto como se estime que el número de puntos que han aparecido en la pantalla es el que se ha solicitado al principio de la partida.

Entonces se obtendrá la visualización de BRAVO o HA PERDIDO, según el resultado obtenido.

EL PROGRAMA: El número de puntos negros se escoge al azar en la línea 1Ø.

El tiempo es descontado por las líneas 4Ø a 9Ø. Los puntos se imprimen por la línea 12Ø, y la línea diagonal se confecciona por la línea 13Ø.

La presión en la tecla Ø se detecta en la línea 15Ø, y la línea 16Ø controla el final de la partida (ningún reflejo).

Las visualizaciones del final se efectúan en las líneas 19Ø a 23Ø.

PARA JUGAR: Se pone en marcha la ejecución con RUN.

Se obtiene entonces la visualización de la frase, EL BUEN REFLEJO ES PARA ... que indica el número de puntos ganadores.

A continuación, se descuenta el tiempo bajo la forma ATENCIÓN ... SEGUNDOS, que indica el tiempo que queda antes de la aparición de la diagonal. A la visualización de 0 aparece la diagonal y empieza a desplazarse sobre la pantalla.

Si se cree que se ha alcanzado el número de puntos reclamados, tendrá que pulsarse la tecla 0. La visualización indica entonces BRAVO, si se ha ganado, o HA PERDIDO en caso contrario; después se visualiza la frase OTRA PRUEBA ¿(S o N)?; responder S para jugar otra partida.

POSIBLES EXTENSIONES: Se considerará una adaptación del juego para dos o más jugadores, dejando que la diagonal se forme completamente y

comprobando la coincidencia en el momento de pulsar la tecla atribuida al jugador.

Puede también realizarse un grafismo más cuidado y modificar el principio del desplazamiento de los objetos en la pantalla (desplazamiento horizontal, vertical o aparición al azar).

DIAGO

PROGRAMA ZX 81

```
   5 CLS
  10 LET X=INT (RND*18+INT PI)
  20 PRINT "EL BUEN REFLEJO ES P
ARA ";X
  30 LET S=10
  40 PRINT "ATENCION :";S;" SEGU
NDOS"
  60 LET S=S-SGN PI
  70 PAUSE 30
  75 CLS
  80 IF NOT S THEN GOTO 100
  90 GOTO 40
 100 LET U=NOT PI
 110 LET V=U
 120 PLOT U,V
 130 SCROLL
 140 LET U=U+SGN PI
 150 IF INKEY$="0" THEN GOTO 180
 160 IF U>21 THEN GOTO 220
 170 GOTO 120
 180 IF U<>X THEN GOTO 220
 190 PRINT "BRAVO ";
 195 PRINT "OTRA PRUEBA (S/N)?"
 200 INPUT R$
 210 IF R$="S" THEN RUN
 215 STOP
 220 PRINT "HA PERDIDO"
 230 GOTO 195
```

```
HA PERDIDO
```

PROGRAMA ZX SPECTRUM

```
   5 CLS
  10 LET X=INT (RND*18+3)
  20 PRINT "EL BUEN REFLEJO ES P
ARA ";X
  30 LET S=10
  40 PRINT FLASH 1;"ATENCION :";
S;" SEGUNDOS"
  60 LET S=S-1
  70 BEEP 1,10: PAUSE 10
  75 CLS
  80 IF NOT S THEN GO TO 100
  90 GO TO 40
 100 LET U=0
 110 LET V=U
 120 CIRCLE 6*U+5,V+5,2
 130 PRINT AT 21,0: POKE 23692,0
: PRINT CHR$ 13
 140 LET U=U+1
 150 IF INKEY$="0" THEN GO TO 18
0
 160 IF U>22 THEN GO TO 220
 170 GO TO 120
 180 IF U<>X THEN GO TO 220
 190 PRINT "BRAVO;";
 195 PRINT " OTRA PRUEBA (S O N)
?"
 200 INPUT R$
 210 IF R$="S" THEN RUN
 215 STOP
 220 PRINT "HA PERDIDO;";
 230 GO TO 195
```

```
BRAVO; OTRA PRUEBA (S O N)?
```

Topología

REFERENCIAS:

Modalidad: FAST o SLOW, impuesta por el programa.
Dificultad de programación: mediana.
Categoría del juego: grafismo.
Dificultad del juego: mediana.

EL JUEGO:

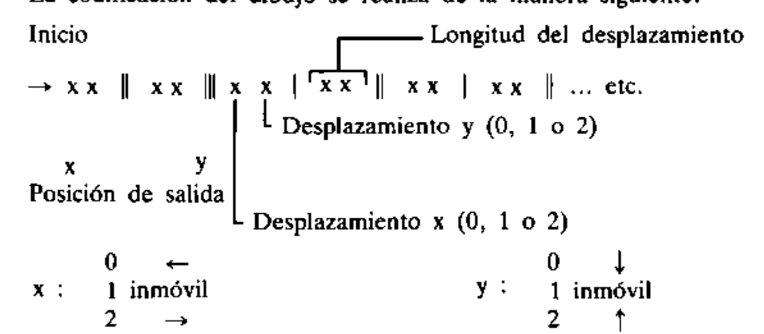
Aparece en la pantalla un dibujo muy elemental a base de trazos (puede también ser introducido por otro jugador).

El jugador dispone de un señalizador móvil, que simboliza una goma de borrar, e intentará suprimir el dibujo anterior repasando, una única vez, sobre cada uno de sus trazos.

EL PROGRAMA:

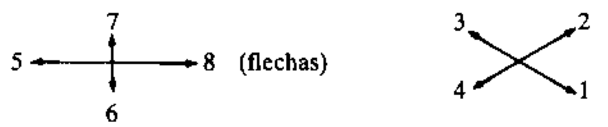
La línea 2Ø contiene un dibujo programado con anterioridad (variable B$). Por supuesto, podrá definirse otro dibujo o introducir el actual pulsando 2Ø INPUT B$.

La codificación del dibujo se realiza de la manera siguiente:



Las líneas 1ØØ y 11Ø calculan la posición de salida.

El bucle I (13Ø-17Ø) realiza el desciframiento, y la impresión está asegurada por el subprograma 2ØØ-23Ø. La parte de borrado empieza en la línea 173. El desciframiento del desplazamiento del jugador está asegurado por las líneas 18Ø y 182 (8 direcciones).



PARA JUGAR:

Se empieza con la instrucción RUN, y si el dibujo no ha sido previamente programado, se le introduce siguiendo las instrucciones dadas con anterioridad. A continuación se pulsa NEWLINE.

Con la visualización del dibujo aparece un punto luminoso intermitente en la parte inferior y a la izquierda de la pantalla. Se puede desplazar utilizando las 4 teclas de desplazamiento o utilizando las teclas 1 a 4 para los desplazamientos en diagonal.

Se intentará borrar el dibujo inicial sin levantar el lápiz y sin repasar dos veces el mismo trazo.

POSIBLES EXTENSIONES: Podrá introducirse color, definir una tabla de dibujos topológicamente interesantes, añadir controles basados en el hecho de no dejar el dibujo inicial, etc.

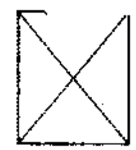
TOPOLOGÍA

PROGRAMA ZX 81

```
  10 FAST
  20 LET B$="12122116121601161O1
6221610160216"
 100 LET S=VAL B$(SGN PI TO 2)
 110 LET T=VAL B$(INT PI TO 4)
 130 FOR I=5 TO LEN B$ STEP 4
 140 LET U=VAL B$(I)-SGN PI
 150 LET V=VAL B$(I+SGN PI)-SGN
PI
 160 GOSUB CODE "COS "
 170 NEXT I
 171 LET X=NOT PI
 172 LET Y=X
 173 SLOW
 174 PLOT X,Y
 176 UNPLOT X,Y
 178 IF INKEY$="" THEN GOTO 174
 180 LET X=X+(INKEY$="8" OR INKE
Y$="1" OR INKEY$="2")-(INKEY$="5
" OR INKEY$="3" OR INKEY$="4")
 182 LET Y=Y+(INKEY$="7" OR INKE
Y$="1" OR INKEY$="4")-(INKEY$="6
" OR INKEY$="2" OR INKEY$="3")
 184 GOTO 174
 200 FOR K=SGN PI TO VAL B$(I+2
TO I+INT PI)
 201 PLOT S,T
 202 LET S=S+U
 204 LET T=T+V
 220 NEXT K
 230 RETURN
```

PROGRAMA ZX SPECTRUM

```
  20 LET B$="32322164126401641O6
4226410640264"
 100 LET S=VAL B$(1 TO 2)
 110 LET T=VAL B$(3 TO 4)
 130 FOR I=5 TO LEN B$ STEP 4
 140 LET U=VAL B$(I)-1
 150 LET V=VAL B$(I+1)-1
 160 GO SUB 200
 170 NEXT I
 171 LET X=150
 172 LET Y=120
 174 PLOT X,Y: PLOT X+1,Y: PLOT
X,Y+1: PLOT X+1,Y+1
 176 PLOT OVER 1;X,Y: PLOT OVER
1;X+1,Y: PLOT OVER 1;X,Y+1: PLOT
 OVER 1;X+1,Y+1
 178 IF INKEY$="" THEN GO TO 174
 180 LET X=X+(INKEY$="8" OR INKE
Y$="1" OR INKEY$="2")-(INKEY$="5
" OR INKEY$="3" OR INKEY$="4")
 182 LET Y=Y+(INKEY$="7" OR INKE
Y$="1" OR INKEY$="4")-(INKEY$="6
" OR INKEY$="2" OR INKEY$="3")
 184 GO TO 174
 200 FOR K=1 TO VAL B$(I+2 TO I+
3)
 201 PLOT S,T
 202 LET S=S+U
 204 LET T=T+V
 220 NEXT K
 230 RETURN
```



Caza al submarino

REFERENCIAS: Modalidad: SLOW o FAST.

Dificultad de programación: mediana.

Categoría del juego: juego de reflexión.

Dificultad del juego: mediana.

EL JUEGO: El jugador está a bordo de un buque de superficie encargado de acosar a un submarino escondido en un cierto lugar de un sector que pertenece al jugador; hay que desplazarse hasta colocarse encima del submarino y disparar a una cierta profundidad; pero cuidado, el submarino también puede moverse; no parará sus motores hasta el momento en que el buque se encuentre exactamente en su vertical, a fin de no ser localizado.

EL PROGRAMA: El bucle I (2Ø-4Ø) visualiza el campo de juego.

Las líneas 97 y 98 definen la posición inicial del submarino; las líneas 1ØØ y 1Ø5 permiten su desplazamiento, controlado por la línea 1Ø7 (hay que quedarse en los límites del campo de juego).

La línea 11Ø visualiza la posición del buque.

Las líneas 117 a 12Ø controlan su desplazamiento.

La distancia se calcula en la línea 13Ø, se visualiza en la línea 14Ø y se comprueba en la línea 15Ø.

En la línea 22Ø se comprueba el final de la partida.

PARA JUGAR: Se pone en marcha el programa por RUN y se introduce la dimensión; se visualiza la rejilla de juego.

El buque se sitúa en la salida, en la posición (0, 0), y hay que desplazarse con las teclas 5, 6, 7, 8.

Después de cada desplazamiento se visualiza la nueva distancia del submarino, que se desplaza a cada disparo. Si consigue situarse en su vertical, se visualiza la distancia 0 y no se mueve más.

Se dispara dando una profundidad entera (entre 2 y 4, límites comprendidos). Si se falla, se desplaza sin rumbo y se vuelve a empezar; si se hunde, se obtiene la visualización BRAVO y la partida termina.

Puede pararse en cualquier momento el programa utilizando la tecla BREAK.

POSIBLES EXTENSIONES: Es posible prever una mejora del gráfico, mejorar las posibilidades de desplazamiento del buque cazador (un poco lento en esta versión), prever trayectos más sofisticados para el submarino, poner un contador de tiempos, etc.

CAZA AL SUBMARINO

PROGRAMA ZX 81

```
  10 LET A=NOT PI
  15 INPUT H
  17 LET G=SGN PI+SGN PI
  20 FOR I=A TO H
  30 PRINT AT I+G,A;I;AT A,G*I+I
NT PI;I
  40 NEXT I
  45 LET I=SGN PI
  75 LET X=A
  78 LET Y=A
  97 LET V=INT (RND*H)
  98 LET U=INT (RND*H)
 100 LET U=U+I
 105 LET V=V+I
 107 IF (U<A OR U>H OR V<A OR V>
H) THEN LET I=-I
 110 PRINT AT X+G,G*Y+SGN PI;"
█  ";AT X+SGN PI,G*Y+INT PI;" ";
AT X+INT PI,G*Y+INT PI;" "
 115 PAUSE PI**H
 117 LET Y=Y+(INKEY$="8" AND Y<H
)-(INKEY$="5" AND Y>A)
 120 LET X=X+(INKEY$="6" AND X<H
)-(INKEY$="7" AND X>A)
 130 LET D=INT SQR (ABS (X-V)**G
+ABS (Y-U)**G)
 140 PRINT AT H+H,A;"DIST :";D;"
 "
 150 IF NOT D THEN GO TO CODE "C
OS "
 180 IF D<INT PI THEN GO TO VAL
"100"
 190 GO TO VAL "110"
 200 PRINT AT H+H,A;"PROF"
 210 INPUT D
 220 IF D=INT (PI*RND)+G THEN PR
INT "BRAVO";W
 230 GO TO VAL "95"
```

```
     0 1 2 3 4 5 6

0
1
2
3
4          █
5
6


DIST :2
```

PROGRAMA ZX SPECTRUM

```
  10 LET A=0
  15 INPUT "DIMENSION :";H
  17 LET G=2
  20 FOR I=A TO H
  30 PRINT PAPER 5;AT I+G,A;I;AT
 A,G*I+3;I
  40 NEXT I
  45 LET I=1
  75 LET X=A
  78 LET Y=A
  97 LET V=INT (RND*H)
  98 LET U=INT (RND*H)
 100 LET U=U+I
 105 LET V=V+I
 107 IF (U<A OR U>H OR V<A OR V>
H) THEN LET I=-I
 110 PRINT INK 2;AT X+G,G*Y+1;"
█  ";AT X+1,G*Y+3;" ";AT X+3,G*
Y+3;" "
 115 PAUSE 4E4
 117 LET Y=Y+(INKEY$="8" AND Y<H
)-(INKEY$="5" AND Y>A)
 120 LET X=X+(INKEY$="6" AND X<H
)-(INKEY$="7" AND X>A)
 130 LET D=INT SQR (ABS (X-V)↑G+
ABS (Y-U)↑G)
 140 PRINT AT H+H,A;"DISTANCIA :
";D;"  "
 150 IF NOT D THEN GO TO 200
 180 IF D<3 THEN BEEP 1,10: GO T
O 100
 190 GO TO 110
 200 PRINT AT H+H,A;"PROFUNDIDAD
:";
 210 INPUT D: PRINT D
 220 IF D=INT (3*RND)+G THEN PRI
NT FLASH 1;"BRAVO": BEEP .3,10:
BEEP .3,18: BEEP .3,8: STOP
 230 GO TO 95
```

```
   0 1 2 3 4 5 6 7 8

0
1
2
3
4
5
6
7
8               █


DISTANCIA :8
```

El lagarto de Alicia

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: fácil.
Categoría del juego: solitario.
Dificultad del juego: bastante difícil.

EL JUEGO: Se inspira muy vagamente en una idea de LEWIS CAROLL, en "Alicia en el País de las Maravillas". En la historia original, Alicia crecía o disminuía de tamaño, mientras que en el juego, por razones de limitación de la memoria, es un lagarto el que cambia de tamaño.

Dos vasos aparecen en la pantalla; uno contiene el líquido que hace crecer, el otro el líquido que hace disminuir (desgraciadamente no se trata siempre del mismo vaso). El jugador deberá indicar la cantidad de líquido que desea beber, y el lagarto cambiará de tamaño en función de esta cantidad.

El objetivo del juego es devolver al lagarto su tamaño original.

EL PROGRAMA: El bucle I (5-35) dibuja los vasos.

Los dos bucles intercalados I (55-65) y J (57-62) los llenan con la cantidad indicada en la línea 5Ø.

La variación del tamaño del lagarto está definida por la línea 75, con control mediante las líneas 8Ø y 9Ø.

El bucle I (1ØØ-3ØØ) dibuja el cuerpo del lagarto, las líneas 5ØØ a 8ØØ dibujan el ojo y las patas.

Un mensaje ganador se visualiza por la línea 81Ø, si se ha encontrado el tamaño normal.

PARA JUGAR: Se pone en marcha el programa con RUN: se dibujan los dos vasos, y después aparece la pregunta VASO, a la cual se responde pulsando I o D; a continuación se obtiene la visualización CUÁNTO; se introduce entonces una cantidad (de 0 a 130) y se pulsa NEWLINE.

El vaso indicado se llena, y después de unos instantes se visualiza el lagarto.

Se pulsará cualquier tecla (excepto BREAK, que terminaría la partida); si el lagarto no tiene el tamaño normal, no se visualizará ningún mensaje y se volverá otra vez al dibujo de los vasos.

POSIBLES EXTENSIONES: Puesto que se trata de un juego esencialmente gráfico, será interesante mejorar los dibujos, introducir color, modificar el carácter aleatorio de la elección de los vasos, etc. ¿Por qué no volver a leer "Alicia en el País de las Maravillas", que está lleno de ideas?

EL LAGARTO DE ALICIA

PROGRAMA ZX 81

```
   2 CLS
   3 LET N=CODE ">"
   4 LET S=SGN PI+SGN PI
   5 FOR I=SGN PI TO 8
  10 PRINT "█    █   █     █"
  20 NEXT I
  30 PRINT "▀▀▀▀▀▀▀▀   ▀▀▀▀▀▀▀▀"
  35 PRINT "IZQUIER  DERECHA"
  38 PRINT AT PI+PI,NOT PI;"VASO
"
  40 IF INKEY$<>"D" AND INKEY$<>
"I" THEN GOTO CODE "C"
  45 LET U=S+CODE ">"*(INKEY$="D
")
  47 PRINT "CUANTO"
  50 INPUT Q
  55 FOR I=NOT PI TO SQR Q
  57 FOR J=NOT PI TO 7
  60 PLOT U+J,I+CODE "4"
  62 NEXT J
  65 NEXT I
  67 PAUSE 4E4
  68 CLS
  70 LET H=2*(RND<.5)-SGN PI
  75 LET N=N+H*Q
  80 LET N=CODE "£"*(N<CODE "£")
+CODE "W"*(N>CODE "W")
 100 FOR I=NOT PI TO N+INT PI
 200 PLOT I,S+INT PI
 300 NEXT I
 400 PLOT N-INT ((SQR N)/S),S+S+
S
 500 PLOT SQR N,S+S
 600 PLOT SQR N+S,S+S
 700 PLOT N-SQR N,S+S
 800 PLOT N-SQR N-S,S+S
 805 PAUSE 4E4
 810 IF N=CODE "0" THEN PRINT "B
RAVO";W
 820 GOTO S
```



IZQUIER DERECHA

VASO
CUANTO

PROGRAMA ZX SPECTRUM

```
   2 INK 0: PAPER 7: CLS
   3 LET N=18
   4 LET S=2
   5 FOR I=1 TO 9
  10 PRINT INK 4;"      █    █   █
█"
  20 NEXT I
  30 PRINT INK 4;"      ▀▀▀▀▀▀▀▀   ▀▀
▀▀▀▀▀▀"
  35 PRINT " IZQUIERDA  DERECHA"
  38 PRINT AT 15,0;"QUE LADO
:";
  40 IF INKEY$<>"D" AND INKEY$<>
"I" THEN GO TO 40
  45 LET U=S+18*(INKEY$="D")
  47 PRINT "DERECHA" AND INKEY$=
"D";"IZQUIERDA" AND INKEY$="I":
PRINT "QUE CANTITAD :";
  50 INPUT Q: PRINT Q
  53 INK 5
  55 FOR I=1 TO 4*SQR Q
  57 FOR J=0 TO 31
  60 PLOT 4*U+J+24,I+103
  62 NEXT J
  65 NEXT I
  67 PAUSE 4E4
  68 BORDER 6: INK 2: CLS
  70 LET H=2*(RND<.5)-1
  75 LET N=N+H*INT SQR Q
  80 IF N<12 THEN LET N=12
  90 IF N>60 THEN LET N=60
 100 FOR I=0 TO 4*(N+3)
 200 PLOT I,4*S+3: PLOT I,4*S+2:
 PLOT I,4*S+1: PLOT I,4*S
 300 NEXT I
 400 LET T=4*(N-INT ((SQR N)/S))
: PLOT T,9*S: PLOT T-1,9*S-1: PL
OT T-2,9*S-1: PLOT T+1,9*S: PLOT
 T+2,9*S-1: CIRCLE T,8*S-2,1.5
 500 LET T=4*SQR N: PLOT T,3*S:
PLOT T-1,3*S-1: PLOT T-2,3*S-2
 600 LET T=T+S: PLOT T,3*S: PLOT
 T+1,3*S-1: PLOT T+2,3*S-2
 700 LET T=4*(N-SQR N): PLOT T,3
*S: PLOT T-1,3*S-1: PLOT T-2,3*S
-2
 800 LET T=T+S: PLOT T,3*S: PLOT
 T+1,3*S-1: PLOT T+2,3*S-2
 805 PAUSE 4E4
 810 IF N=28 THEN PRINT "BRAVO":
 STOP
 820 GO TO 2
```



IZQUIERDA DERECHA

QUE LADO   :DERECHA
QUE CANTITAD :130

BRAVO

Calculadora

REFERENCIAS:

Modalidad: FAST o SLOW.

Dificultad de programación: muy difícil.

Categoría del juego: simulación de una calculadora.

Dificultad del juego: mediana.

EL JUEGO:

Se trata de la simulación, en una visualización de gran tamaño, del funcionamiento de una pequeña calculadora de bolsillo, del tipo de 4 operaciones. Las aplicaciones pedagógicas de este programa son evidentes, y su funcionamiento es muy simple de dominar.

EL PROGRAMA:

Las variables B$ y C$ de las líneas 2Ø y 3Ø contienen las informaciones codificadas, que permitirán el dibujo de las cifras en tamaño grande.

Las líneas 129 a 15Ø comprueban las distintas entradas posibles.

Las líneas 16Ø a 18Ø se ocupan de la representación en tamaño grande de las cifras, a medida que se introducen en la pantalla.

El punto decimal y el signo – son gestionados por la línea 185.

El número total de símbolos introducidos se controla por la línea 19Ø.

El resultado se inscribe gracias a un programa que funciona automáticamente, y que se encuentra en las líneas 229 a 27Ø (el automatismo se gestiona en el programa principal por la línea 195).

PARA JUGAR:

Se pondrá en marcha el programa por RUN.

Después se pulsan las cifras del primer número que se introduce y que van a visualizarse, una a una, a medida que se teclean (no se podrán utilizar más de 7 símbolos, comprendidos el signo – y el punto decimal).

La pulsación de un signo operacional deja la visualización intacta; pero la introducción de la siguiente cifra o símbolo provoca la supresión inmediata y la visualización del nuevo símbolo. Hay que advertir que las cifras se introducen de izquierda a derecha, en el mismo sentido que la escritura normal.

El aparato acepta como símbolos operacionales los siguientes: +, –, * y /; el signo = provocará la visualización del resultado. Las operaciones son automáticamente encadenadas; el resultado anterior se tiene en cuenta como operador del siguiente cálculo.

Para realizar otra operación se pulsará BREAK y después RUN.

Pulsar BREAK para salir del programa.

POSIBLES EXTENSIONES: Ante todo, deberá mejorarse la visualización del signo – (representado aquí por un punto); después podrá mejorarse el dibujo de las cifras (utilizando, por ejemplo, el generador del ZX). También podrá considerarse una ampliación a 8 cifras, autorizar operaciones más complejas (COS, SEN), incorporar memorias, etc.

CALCULADORA

PROGRAMA ZX 81

```
   5 LET U=SGN PI+SGN PI
  30 LET C$="███   █ ▌"
  40 LET B$="1444133333131211313
14413312131121411333314141141 31"
  45 LET R$=""
  50 LET T=NOT PI
  60 LET N=NOT PI
 128 PAUSE 4E4
 129 IF INKEY$="■" THEN GOTO COD
E " FAST "
 130 LET R$=R$+INKEY$
 132 IF INKEY$<"." OR INKEY$>"9"
 THEN GOTO CODE "W"
 135 LET P$=INKEY$
 145 IF NOT N THEN CLS
 150 IF P$="." OR P$="-" THEN GO
TO CODE "█"
 160 FOR I=SGN PI TO INT PI+U
 165 LET S=U*VAL B$((INT PI+U)*V
AL P$+I)-SGN PI
 170 PRINT AT I,N;C$(S TO S+SGN
PI)
 180 NEXT I
 185 PLOT N+N+INT PI,CODE "4"
 190 LET N=N+(U+U)*(N<CODE "/")
 195 IF T THEN NEXT T
 200 GOTO CODE "█"
 229 LET N=NOT PI
 230 LET R$=STR$ VAL R$
 250 FOR T=SGN PI TO LEN R$
 260 LET P$=R$(T)
 270 GOTO CODE "█"
```

435.678

PROGRAMA ZX SPECTRUM

```
  10 LET U=2
  20 LET C$="███   █ ▌"
  30 LET B$="1444133333131211313
14413312131121411333314141141 31"
  40 LET R$=""
  50 LET T=0
  60 LET N=0
  70 LET H=0
 128 PAUSE 4E4
 129 IF INKEY$="=" THEN GO TO 22
9
 130 LET R$=R$+INKEY$
 131 IF R$="-" THEN LET P$=R$: G
O TO 185
 132 IF INKEY$<"." OR INKEY$>"9"
 OR INKEY$="/" THEN GO TO 60
 135 LET P$=INKEY$
 145 IF NOT N THEN CLS
 150 IF P$="." OR P$="-" THEN GO
 TO 185
 160 FOR I=1 TO 5
 165 LET S=U*VAL B$(I+5*VAL P$)-
1
 170 PRINT INK 2-H;AT I,N;C$(S T
O S+1)
 180 NEXT I
 185 IF P$="-" THEN PRINT INK 2-
H;AT 3,2;"▀"
 187 IF P$="." THEN PRINT INK 2-
H;AT 3,N;"▄"
 188 BEEP .3,20
 190 LET N=N+4*(N<24)
 195 IF T THEN NEXT T
 200 GO TO 128
 229 LET N=0
 230 LET R$=STR$ VAL R$
 232 LET H=NOT H
 240 BEEP 1,-5
 250 FOR T=1 TO LEN R$
 260 LET P$=R$(T)
 270 GO TO 145
```

1234·56

NIVEL 5

N.°	*Nombre del programa*	*Tema y particularidades*
1	**Smokey**	Fumar sin riesgo, con este original programa para un jugador.
2	**Carreras de coches**	Un juego de reflejos, no muy difícil, para un jugador.
3	**Ajedrez**	Un programa muy simplificado, para dos jugadores.
4	**Peligro espacial**	¿Será suficientemente hábil para evitar todos los meteoritos?
5	**Tic tac toe**	Un clásico, pero no por ello sin interés (un jugador).
6	**Reloj 2**	Utilización del generador de caracteres incorporado.
7	**Damas**	Para dos jugadores; clásico.
8	**Anagramas**	Un buen juego de reflejos, para dos jugadores.
9	**Huevos**	¡No dejarse manejar por la ponedora diabólica! (un jugador).
10	**Visualización giratoria (vertical)**	Una posibilidad de aprovecharse del ZX.
11	**De cuatro en cuatro**	Un juego de reflexión, un poco difícil, para dos jugadores.
12	**Revoltijo de letras**	Un proverbio se esconde dentro; es su ocasión para descubrirlo.
13	**Blancos**	Un poco de habilidad, mucha suerte. Muy visual.
14	**Gotas**	Más bien un efecto visual y sonoro. Para utilizar.
15	**Obstáculos**	Muchos obstáculos hay que evitar en este juego de habilidad para un jugador.
16	**Capturas en el espacio**	¡Una misión fácil, sin los dichosos meteoritos!
17	**Anuncio publicitario (2)**	¡Inclinar el televisor unos 90° y todo irá bien!
18	**Anomalía**	Para ejercer la perspicacia en el reconocimiento de los símbolos.
19	**Varios en uno**	¡Un terreno de juego y peones, para dos jugadores! ¡Escojan!
20	**Recorrido del caballo**	Muy espectacular e instructivo por varias razones.
21	**Pared de ladrillos**	Un gran clásico, en una versión muy completa.
22	**Othello**	Para dos jugadores; le dará la vuelta a los peones en su lugar.

Inscripción directa en las memorias

LA INSTRUCCIÓN DE EXPLORACIÓN DE LAS POSICIONES DE MEMORIA: PEEK

Esta **instrucción** (utilizada con la instrucción complementaria **POKE**) **permite leer** (y escribir) **directamente en cualquiera de las posiciones de memoria direccionables por el microprocesador**; así, utilizando con cuidado los **POKE** y los **PEEK**, podrán remediarse las insuficiencias de lenguaje Basic, llegando incluso hasta la escritura, a partir del Basic, de programas en lenguaje máquina.

En lo que nos concierne, hemos utilizado el POKE en el ZX Spectrum para inhibir la visualización SCROLL ? , que aparece automáticamente en la parte inferior de la pantalla cuando ésta se encuentra llena. La dirección de memoria correspondiente es 23692, y bastará con escribir POKE 23692, Ø para evitar esta visualización intempestiva.

En el ZX 81, la función PEEK se utiliza sobre todo para localizar una dirección de la memoria en la zona de visualización: en efecto, las posiciones de memoria 16398 y 16399 contienen en dos octetos la dirección de la siguiente posición de impresión (gestión del PRINT); se utiliza esta propiedad para mirar lo que se encuentra en un lugar señalado de la pantalla.

Se escribe PRINT AT X,Y ; lo que prepara al ZX para escribir en la línea X, columna Y.

Se lee entonces la dirección correspondiente a esta posición mediante PEEK 16398 + 256 × PEEK 16399; para saber lo que se encuentra en la pantalla, bastará con hacer un PEEK de esta dirección: se encuentra un número de Ø a 255, que corresponde al código del carácter presente en la memoria de la pantalla.

Se procede de la misma forma en el ZX Spectrum, pero será más simple utilizar una de las dos funciones preprogramadas SCREEN$ (lectura de un carácter en la pantalla) y ATTR(X,Y) (lectura de los atributos de color del punto X,Y de la pantalla).

Se utilizará igualmente la función PEEK para leer las informaciones contenidas en la memoria inactiva, por ejemplo, el generador de caracteres, situado en ROM a partir de la dirección 7680 en el ZX 81 y 15360 en el ZX Spectrum.

Smokey

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: solitario.

Dificultad del juego: mediana.

EL JUEGO: Una columna de humo gris aparece en la pantalla. El jugador, representado por un punto negro, es visible a la derecha. En un primer momento debe colocarse sobre la columna de humo, entonces, desde allí va a tragarse automáticamente el humo a condición de que consiga mantenerse en su trazado.

EL PROGRAMA: La línea 3Ø hace aparecer el humo en la parte baja de la pantalla; la línea 35 permite cambiar (hacia la derecha o hacia la izquierda) la dirección de la columna, y la línea 5Ø hace subir la columna.

El jugador aparece gracias a la línea 66, y su desplazamiento se efectúa por las líneas 9Ø y 1ØØ.

Las líneas 12Ø y 19Ø se ocupan de la visualización del resultado.

PARA JUGAR: Debe ponerse en marcha el programa por RUN.

El humo empieza a subir, mientras que el jugador aparece en forma de un cuadrado negro en una esquina de la pantalla; se desplazará sobre ella con la ayuda de 4 flechas. Tan pronto como llegue el humo sobre el jugador, éste empezará a tragárselo automáticamente e intentará seguir la columna sin desviarse (gracias a las flechas 8 y 5).

El jugador ganará si llega abajo de la pantalla antes de que el humo llegue arriba.

La visualización final indica en este momento BRAVO o HA PERDIDO.

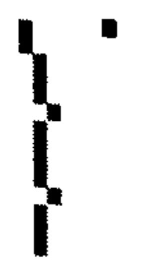
POSIBLES EXTENSIONES: Podrán, conservando la idea inicial, imponerse otras reglas: por ejemplo, tragar todo el humo, que podrá también salir de otro sitio si se le desvía de la columna inicial.

Podrán mejorarse los controles del final de la partida y la calidad del grafismo.

SMOKEY

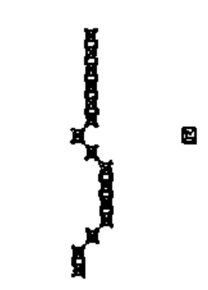
PROGRAMA ZX 81

```
  10 LET X=CODE "="
  15 LET U=NOT PI
  20 LET V=CODE "£"
  25 LET Y=INT (CODE ")"*RND)
  30 PRINT AT X,Y;"▒"
  35 LET Y=Y+(RND<.5 AND Y<CODE
")")-(RND<.5 AND Y>NOT PI)
  40 LET T=SGN PI
  50 LET X=X-T
  60 IF NOT X THEN GOTO CODE "■"
  65 PRINT AT U,V;"■"
  70 IF T THEN PAUSE CODE "PI"
  80 PRINT AT U,V;" "
  90 LET U=U+(INKEY$="6" AND T A
ND X<CODE "=")-(INKEY$="7" AND T
 AND X>NOT PI)
  95 LET U=U+NOT T
 100 LET V=V+(INKEY$="8" AND V<C
ODE ")")-(INKEY$="5" AND V>NOT P
I)
 110 PRINT AT U,V;
 120 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
 130 IF CHR$ C<>"▒" THEN GOTO CO
DE "2"
 140 LET T=NOT PI
 160 IF U<>CODE "=" THEN GOTO CO
DE "PI"
 180 CLS
 190 PRINT "BRAVO" AND NOT T;"HA
 PERDIDO" AND T
```



PROGRAMA ZX SPECTRUM

```
   5 BORDER 6: LET S=0
  10 LET X=20
  15 LET U=0
  20 LET V=12
  25 LET Y=INT (17*RND)
  30 PRINT INVERSE 1; INK 4;AT X
,Y;"©"
  35 LET Y=Y+(RND<.5 AND Y<17)-(
RND<.5 AND Y>0)
  40 LET T=1
  50 LET X=X-T
  60 IF NOT X THEN GO TO 180
  65 PRINT FLASH 1; INVERSE 1; I
NK 2;AT U,V;"●"
  70 IF T THEN PAUSE 66
  80 PRINT AT U,V;" "
  90 LET U=U+(INKEY$="6" AND T A
ND X<20)-(INKEY$="7" AND T AND X
>0)
  95 LET U=U+NOT T
 100 LET V=V+(INKEY$="8" AND V<1
7)-(INKEY$="5" AND V>0)
 120 LET C$=SCREEN$ (U,V)
 130 IF C$<>"©" THEN GO TO 30
 140 LET T=0: BEEP .05,0: LET S=
S+1
 160 IF U<>20 THEN GO TO 65
 180 CLS
 190 PRINT "BRAVO" AND NOT T;"HA
 PERDIDO" AND T;" MARCADOR :"; F
LASH 1;S: BEEP .3,10: BEEP .3,10
: BEEP .3,6
```



Carreras de coches

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: juego de reflexión (un jugador).
Dificultad del juego: mediana.

EL JUEGO: Se circula por una carretera muy sinuosa en la que no hay ningún obstáculo, y se intentará seguirla el máximo de tiempo posible.

El recorrido queda visualizado permanentemente, y la puntuación se muestra en cuanto ocurre un accidente.

EL PROGRAMA: La impresión de la carretera se asegura por la línea 5Ø.

En la línea 55, la instrucción SCROLL hace adelantar la carretera, y la línea 6Ø imprime el coche.

La línea 8Ø controla su movimiento, y la línea 9Ø determina la existencia de curvas o no en la carretera.

Las líneas 1ØØ y 11Ø miran si se tocan los bordes de la carretera.

La línea 12Ø aumenta la puntuación, y la línea 13Ø borra la antigua posición del coche (un trozo claro aparece en el lugar de su trayectoria).

PARA JUGAR: Poner en marcha la ejecución del programa por la orden RUN; el dibujo de la carretera empieza a seguir su curso y aparece el vehículo, representado por una Y invertida.

Para desplazarlo a la derecha o a la izquierda, se utilizarán las teclas 5 y 8 (flechas de desplazamiento).

Se evitará, el máximo de tiempo posible, tocar un reborde negro, lo que provocará la parada del programa y la visualización de la mención ACCIDENTE, seguida de la indicación sobre la puntuación realizada.

Podrá pararse en cualquier momento el desarrollo de la partida pulsando la tecla BREAK.

POSIBLES EXTENSIONES: Se considerará el juego para varias personas, grabando la mejor puntuación.

Puede también considerarse el ensanchamiento de la pista y hacer correr a varios coches.

También podrá inspirarse en los juegos de arcadas, los cuales son particularmente numerosos sobre este tema.

CARRERAS DE COCHES

PROGRAMA ZX 81

```
  10 LET A=INT PI*INT PI
  20 LET B=A
  30 LET X=A+A
  40 LET Y=A-INT PI
  45 LET S=NOT PI
  50 PRINT AT X,Y;"██...██"
  55 SCROLL
  60 PRINT AT A,B;"█"
  65 LET C=B
  80 LET B=B+(INKEY$="8")-(INKEY
$="5")
  90 LET Y=Y+(2*RND AND Y<CODE "
(")-(2*RND AND Y>INT PI)
 100 PRINT AT A+SGN PI,B;
 110 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "█" THEN PRINT "AC
CIDENTE ,MARCADOR ";S;W
 120 LET S=S+SGN PI
 130 PRINT AT A,C;" "
 140 GOTO CODE "M"
```

ACCIDENTE ,MARCADOR 18

PROGRAMA ZX SPECTRUM

```
  10 LET A=9
  20 LET B=A
  30 LET X=A+A
  40 LET Y=A-3
  45 LET S=0
  50 PRINT INK 1;AT X,Y;"██...██
"
  55 PRINT AT 20,0: POKE 23692,0
: PRINT CHR$ 13
  60 PRINT INK 4;AT A,B;"█"
  65 LET C=B
  80 LET B=B+(INKEY$="8")-(INKEY
$="5")
  90 LET Y=Y+(2*RND AND Y<16)-(2
*RND AND Y>3)
 100 PAUSE 6
 110 IF SCREEN$ (A+1,B)="*" THEN
 BEEP .2,10: BEEP .2,5: BEEP .2,
10: PRINT INK 2; FLASH 1;"ACCIDE
NTE"; INK 0; FLASH 0;" MARCADOR
:";S;W
 120 LET S=S+1
 130 PRINT AT A,C;" "
 140 GO TO 50
```



Ajedrez

REFERENCIAS:

Modalidad: SLOW o FAST.

Dificultad de programación: difícil.

Categoría del juego: para dos jugadores.

Dificultad del juego: difícil.

EL JUEGO:

No podía pensarse que en 1K-Octeto y utilizando el Basic pudiera realizarse un programa que permitiera al ZX jugar al ajedrez

El programa que se propone permitirá solamente a dos jugadores disputar una partida, utilizando la pantalla del televisor como tablero de juego.

No hay control de validez de las jugadas; solamente una gestión de las piezas sobre el tablero. Las piezas se representan por su inicial, P para los peones, T para las torres, A para los alfiles, C para los caballos, D para las reinas y R para los reyes. El color del fondo permite distinguir las piezas negras de las piezas blancas.

EL PROGRAMA:

La línea 1Ø realiza la impresión del tablero.

La línea 2Ø indica el jugador que tiene el turno (negras o blancas).

Las líneas 5Ø y 6Ø detectan el símbolo presente en la casilla de salida.

La línea 7Ø verifica que la casilla designada no esté vacía y que la pieza que se maneja esté en su color correspondiente.

La línea 8Ø busca si la pieza que tiene que desplazarse ocupaba una casilla blanca o una casilla negra, y la línea 9Ø saca la pieza, volviendo a imprimir el color de la casilla.

Las líneas 1ØØ y 12Ø solicitan la casilla de llegada.

La línea 13Ø realiza la impresión.

La línea 15Ø asegura el cambio de jugador.

PARA JUGAR:

Se pone en marcha la ejecución por RUN.

Se dibuja el tablero y aparece la pregunta JR BLANCAS SALIDA (LC) ; se introduce la línea, después la columna de la pieza que se moverá, bajo la forma, por ejemplo : 24 (peón blanco de la línea 2 y de la columna 4), seguido de NEWLINE. La pieza desaparece del tablero y se obtiene la visualización P : 24 (notificación de la pieza desplazada) JR BLANCAS LLEGADA (LC) ; entonces se teclea la línea y la columna de la casilla de llegada, bajo la forma, por ejemplo : 44 , seguido de NEWLINE.

POSIBLES EXTENSIONES: Podrá realizarse un grafismo más preciso (un posible ejemplo se proporciona en esta misma obra). Sin demasiadas dificultades, podrá también introducirse el control del movimiento de las piezas. No quedará nada más que escribir el programa del juego del ZX ¡ÁNIMO!

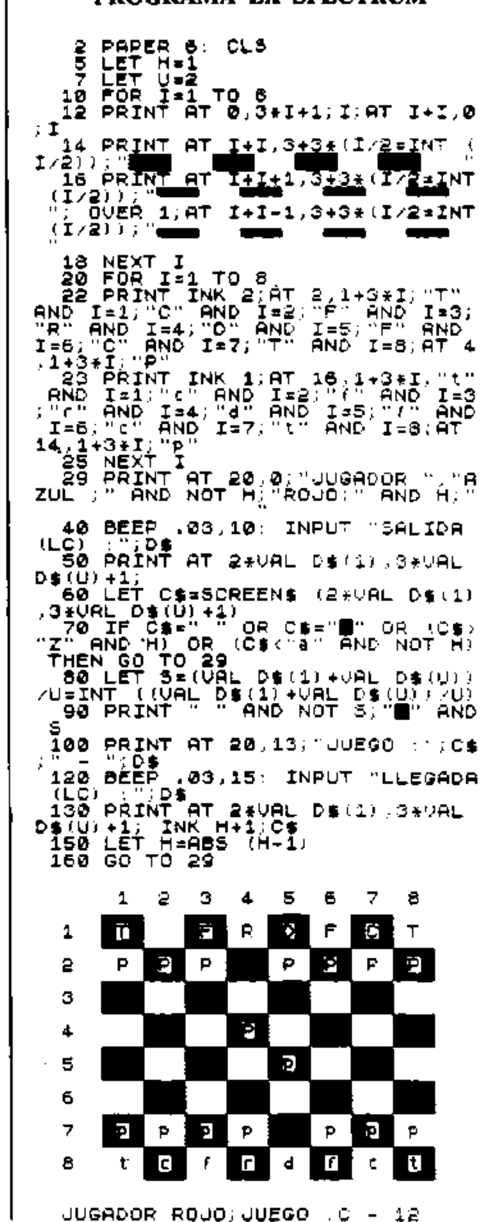
AJEDREZ

PROGRAMA ZX 81

```
   5 LET H=SGN PI
   7 LET U=H+H
  10 PRINT " 12345678",,"1TCFRDF
CT",,"2PPPPPPPP",,"3█ █ █ █",,"4
█ █ █ █",,"5█ █ █ █",,"6█ █ █ █
█",,"7█ █ █ █",,"8█ █ █ █"
  20 PRINT AT CODE "2",NOT PI;"J
R ";"BLANCAS" AND H;"NEGRAS " AN
D NOT H
  30 PRINT "SAL. (LC)"
  40 INPUT D$
  50 PRINT AT VAL D$(SGN PI),VAL
 D$(U);
  60 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
  70 IF C=CODE " " OR (CHR$ C>="
█" AND H) OR (CHR$ C<="█" AND NO
T H) THEN GOTO CODE "="
  80 LET S=(VAL D$(SGN PI)+VAL D
$(U))/U=INT ((VAL D$(SGN PI)+VAL
 D$(U))/U)
  90 PRINT " " AND NOT S;"█" AND
 S
 100 PRINT AT PI+PI,NOT PI;CHR$
C;":";D$;AT CODE "$",NOT PI;"LLE
G. (LC)"
 120 INPUT D$
 130 PRINT AT VAL D$(SGN PI),VAL
 D$(U);CHR$ C;AT PI+PI,NOT PI;"
           "
 150 LET H=ABS (H-SGN PI)
 160 GOTO CODE "="
```



PROGRAMA ZX SPECTRUM



```
   2 PAPER 6: CLS
   5 LET H=1
   7 LET U=2
  10 FOR I=1 TO 8
  12 PRINT AT 0,3*I+1;I;AT I+I,0
;I
  14 PRINT AT I+I,3+3*(I/2=INT (
I/2));"███   ███   ███   ███    "
  16 PRINT AT I+I+1,3+3*(I/2=INT
 (I/2));"▄▄▄   ▄▄▄   ▄▄▄   ▄▄▄
"; OVER 1;AT I+I-1,3+3*(I/2=INT
 (I/2));"▄▄▄   ▄▄▄   ▄▄▄   ▄▄▄
"
  18 NEXT I
  20 FOR I=1 TO 8
  22 PRINT INK 2;AT 2,1+3*I;"T"
AND I=1;"C" AND I=2;"F" AND I=3;
"R" AND I=4;"D" AND I=5;"F" AND
I=6;"C" AND I=7;"T" AND I=8;AT 4
,1+3*I;"P"
  23 PRINT INK 1;AT 16,1+3*I,"t"
 AND I=1;"c" AND I=2;"f" AND I=3
;"r" AND I=4;"d" AND I=5;"f" AND
 I=6;"c" AND I=7;"t" AND I=8;AT
14,1+3*I;"p"
  25 NEXT I
  29 PRINT AT 20,0;"JUGADOR ","A
ZUL ;" AND NOT H;"ROJO:" AND H;"
 "
  40 BEEP .03,10: INPUT "SALIDA
(LC) :";D$
  50 PRINT AT 2*VAL D$(1),3*VAL
D$(U)+1;
  60 LET C$=SCREEN$ (2*VAL D$(1)
,3*VAL D$(U)+1)
  70 IF C$=" " OR C$="█" OR (C$>
"Z" AND H) OR (C$<"a" AND NOT H)
 THEN GO TO 29
  80 LET S=(VAL D$(1)+VAL D$(U))
/U=INT ((VAL D$(1)+VAL D$(U))/U)
  90 PRINT " " AND NOT S;"█" AND
 S
 100 PRINT AT 20,13;"JUEGO :";C$
;" - ";D$
 120 BEEP .03,15: INPUT "LLEGADA
 (LC) :";D$
 130 PRINT AT 2*VAL D$(1),3*VAL
D$(U)+1; INK H+1;C$
 150 LET H=ABS (H-1)
 160 GO TO 29
```

Peligro espacial

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: juego de reflejos, para un jugador.
Dificultad del juego: mediana.

EL JUEGO: El jugador está al mando de un vehículo espacial, que acaba de penetrar en una nube de meteoritos.
Se intentará atravesar la nube, sin chocar con ningún meteorito.
La puntuación se visualizará al final de la partida, o cuando se haya colisionado con uno de ellos.

EL PROGRAMA: La posición de cada meteorito está determinada por la línea 1Ø, y se imprimen por la línea 2Ø (gráfico de desplazamiento A).
El avance de los meteoritos se realiza por dos instrucciones SCROLL, en las líneas 3Ø y 4Ø.
Las líneas 45, 46 y 47 verifican si se ha chocado con un meteorito.
La línea 6Ø dibuja la nave espacial, y la línea 7Ø permite desplazarla.
El contador se encuentra en la línea 8Ø.

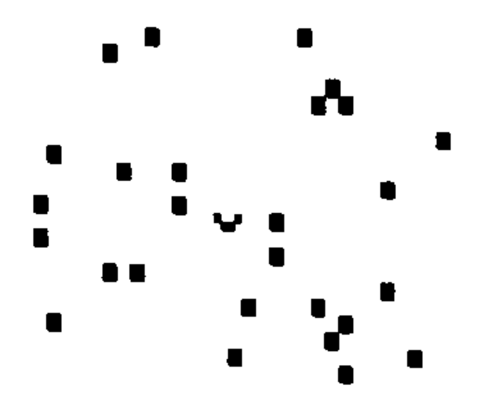
PARA JUGAR: El juego se pone en marcha por la orden RUN.
La nave espacial aparece en el centro de la pantalla, y los meteoritos salen a su encuentro, partiendo de abajo de la pantalla.
Para desplazarse, pueden utilizarse las teclas de desplazamiento horizontal (5 y 8).
Puede pararse la partida en cualquier momento usando BREAK. Si no, el juego termina cuando se choca con un meteorito; entonces se visualiza la puntuación.

POSIBLES EXTENSIONES: Sería interesante que el juego fuera cada vez más difícil, haciendo aparecer más meteoritos a medida que se desarrolla la partida.
También puede intentarse destruir los meteoritos pulverizándolos a medida que vayan apareciendo.

PELIGRO ESPACIAL

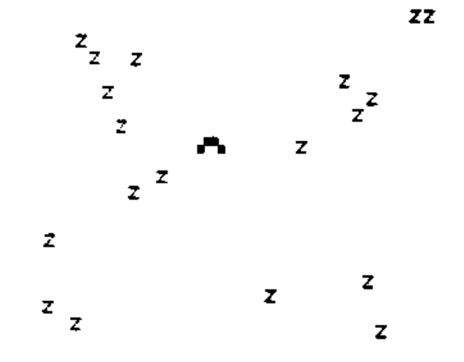
PROGRAMA ZX 81

```
   5 LET X=12
   7 LET Z=NOT PI
  10 LET H=INT (RND*30)
  20 PRINT AT 20,H;"█" AND RND<.
7
  30 SCROLL
  40 SCROLL
  45 PRINT AT 12,X;
  46 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
  47 IF CHR$ C="█" THEN GO TO CO
DE "COS "
  50 PRINT AT 10,X-2;"       "
  60 PRINT AT 12,X;"▙▟"
  70 LET X=X+(INKEY$="8" AND X<3
0)-(INKEY$="5" AND X>NOT PI)
  80 LET Z=Z+SGN PI
  90 GO TO 10
 200 CLS
 210 PRINT "MARCADOR ",Z
```



PROGRAMA ZX SPECTRUM

```
   5 LET X=12
   7 LET Z=0
  10 LET H=INT (RND*30)
  20 PRINT PAPER 2;AT 20,H;"Z" A
ND RND<.7
  30 PRINT PAPER 6;AT 20,INT (RN
D*30);"Z" AND RND<.6
  40 POKE 23692,0: PRINT CHR$ 13
  45 LET B$=SCREEN$ (12,X+1)
  46 LET C$=SCREEN$ (12,X)
  47 IF C$="Z" OR B$="Z" THEN GO
 TO 200
  50 PRINT AT 10,X-2;"       "
  60 PRINT INK 5;AT 11,X;"▙▟"
  70 LET X=X+(INKEY$="8" AND X<3
0)-(INKEY$="5" AND X>0)
  80 LET Z=Z+1
  90 GO TO 10
 200 CLS
 210 PRINT FLASH 1;"MARCADOR :";
 FLASH 0;Z: BEEP .3,10: BEEP .3,
10: BEEP .3,8
```



Tic tac toe

REFERENCIAS: Modalidad: SLOW.

Dificultad de programación: mediana.

Categoría del juego: de reflexión (dos jugadores).

Dificultad del juego: fácil.

EL JUEGO: Un gran clásico de los juegos para escolares, propuesto aquí en una versión que no implica el juego de la máquina; cada jugador, en su turno, deberá colocar un Ø o una X en una casilla libre de la rejilla del juego, y el primero que consiga un alineamiento de 3 símbolos idénticos ha ganado (el alineamiento puede ser horizontal, vertical o siguiendo una de las dos diagonales).

EL PROGRAMA: El campo de juego se dibuja con la ayuda de dos bucles I (1Ø-25 y 27-4Ø).

La línea 6Ø se ocupa de visualizar el símbolo X o Ø, correspondiente al jugador que tiene el turno.

Un símbolo intermitente (línea 11Ø) se visualiza arriba y a la izquierda del campo, incluso si la casilla ya está ocupada; las líneas 123 y 125 permiten desplazar el símbolo sobre el campo de juego.

La línea 14Ø verifica si la casilla que se quiere jugar está libre y si se desea utilizarla.

La línea 15Ø visualiza la jugada, y la línea 16Ø realiza el cambio de jugador.

PARA JUGAR: En cuanto se pulsa RUN se visualiza el campo de juego.

Un símbolo intermitente aparece entonces arriba y a la izquierda de la pantalla; el jugador puede desplazarlo sobre el campo de juego con la ayuda de 4 flechas (teclas 5, 6, 7, 8).

Cuando se llega sobre la casilla que se quiere jugar, basta con pulsar la tecla Ø; la jugada será rechazada si la casilla no está libre.

Si la casilla está libre, se inscribirá el símbolo que corresponde a este jugador y se realizará el cambio de jugador.

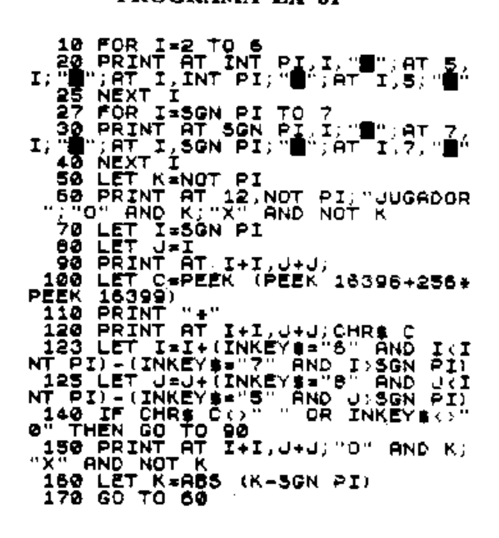
Cuando se termina la partida, será conveniente pulsar BREAK para salir del programa.

A continuación, podrá jugarse otra partida con RUN.

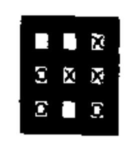
POSIBLES EXTENSIONES: Después de cada jugada podría realizarse un control para verificar si uno de los jugadores ha ganado. También puede realizarse un juego de máquina, pero debería evitarse que fuera demasiado sistemático (por ejemplo, la máquina jugando siempre su primera jugada sobre la casilla central).

TIC TAC TOE

PROGRAMA ZX 81



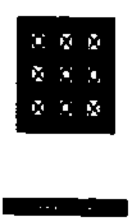
```
  10 FOR I=2 TO 6
  20 PRINT AT INT PI,I;"█";AT 5,
I;"█";AT I,INT PI;"█";AT I,5;"█"
  25 NEXT I
  27 FOR I=SGN PI TO 7
  30 PRINT AT SGN PI,I;"█";AT 7,
I;"█";AT I,SGN PI;"█";AT I,7;"█"
  40 NEXT I
  50 LET K=NOT PI
  60 PRINT AT 12,NOT PI;"JUGADOR
";"O" AND K;"X" AND NOT K
  70 LET I=SGN PI
  80 LET J=I
  90 PRINT AT I+I,J+J;
 100 LET C=PEEK (PEEK 16396+256*
PEEK 16399)
 110 PRINT "*"
 120 PRINT AT I+I,J+J;CHR$ C
 123 LET I=I+(INKEY$="6" AND I<I
NT PI)-(INKEY$="7" AND I>SGN PI)
 125 LET J=J+(INKEY$="8" AND J<I
NT PI)-(INKEY$="5" AND J>SGN PI)
 140 IF CHR$ C<>" " OR INKEY$<>"
0" THEN GO TO 90
 150 PRINT AT I+I,J+J;"O" AND K;
"X" AND NOT K
 160 LET K=ABS (K-SGN PI)
 170 GO TO 60
```



JUGADOR X

PROGRAMA ZX SPECTRUM

```
  10 FOR I=2 TO 6
  20 PRINT INK 5;AT 3,I;"█";AT 5
,I;"█";AT I,3;"█";AT I,5;"█"
  25 NEXT I
  27 FOR I=1 TO 7
  30 PRINT INK 1;AT 1,I;"█";AT 7
,I;"█";AT I,1;"█";AT I,7;"█"
  40 NEXT I
  50 LET K=0
  60 PRINT PAPER 4; INVERSE 1;AT
 12,0;"JUGADOR ";"O" AND K;"X" A
ND NOT K
  70 LET I=1
  80 LET J=I
  90 PRINT AT I+I,J+J;
 100 LET C=CODE SCREEN$ (I+I,J+J
)
 110 PRINT "*"
 120 PRINT AT I+I,J+J;CHR$ C
 123 LET I=I+(INKEY$="6" AND I<3
)-(INKEY$="7" AND I>1)
 125 LET J=J+(INKEY$="8" AND J<3
)-(INKEY$="5" AND J>1)
 140 IF CHR$ C<>" " OR INKEY$<>"
0" THEN GO TO 90
 150 PRINT AT I+I,J+J;"O" AND K;
"X" AND NOT K: BEEP .3,10+8*K
 160 LET K=ABS (K-1)
 170 GO TO 60
```



Reloj 2

REFERENCIAS: Modalidad: FAST.
Dificultad de programación: difícil.
Categoría del juego: dibujo.
Dificultad del juego: ninguna.

EL JUEGO: Este segundo programa de simulación de un reloj digital en tamaño grande presenta la interesante particularidad de utilizar el generador de caracteres del ZX.

Así, son los caracteres de origen los que se reproducirán ampliados.

EL PROGRAMA: La línea 11Ø permite la introducción de la hora, bajo la forma de un entero de 4 cifras.

La línea 13Ø recurre al generador, el cual dibuja los caracteres ampliados (subprograma clásico de la línea 9ØØØ a la línea 915Ø).

Los caracteres son establecidos a partir de la dirección 768Ø en el ZX 81.

La línea 22Ø se encarga de la puesta a cero al cabo de 24 horas.

Las líneas 17Ø, 173 y 175 calibran el número introducido, para realizar una visualización en tamaño estándar.

El ajuste del tiempo se realiza en la línea 14Ø.

PARA JUGAR: Pulsar RUN, y después introducir un entero de 4 cifras (las dos primeras representan la hora, las dos siguientes los minutos : Ø416 significará 4 horas y 16 minutos).

Esperar la cuarta señal para pulsar NEWLINE, y empezará la visualización.

Podrá ajustarse el reloj aprovechando la duración de la pausa, en la línea 14Ø.

POSIBLES EXTENSIONES: Puede hacerse que los dos puntos sean intermitentes, añadir la posibilidad del reloj despertador, prever una agenda automática, hacer aparecer un cuco cantando al sonar las horas, etc.

RELOJ 2

PROGRAMA ZX 81

```
 100 INPUT A$
 110 CLS
 115 LET U=CODE "£"
 120 LET V=CODE ","
 130 GOSUB 9000
 140 PAUSE 2700
 150 LET I=SGN PI+VAL A$(4 TO )
 155 LET J=VAL A$( TO 2)
 160 IF I=CODE "U" THEN GOSUB CO
DE "COS "
 170 LET A$="0"+STR$ (J)
 173 LET A$=A$(LEN A$-SGN PI TO
)+":"+"0"+STR$ (I)
 175 LET A$=A$( TO INT PI)+A$(LE
N A$-SGN PI TO )
 180 GOTO 110
 200 LET I=NOT PI
 210 LET J=J+SGN PI
 220 IF J=CODE "/" THEN LET J=NO
T PI
 230 RETURN
9000 LET Z$="█RND4([illegible] [illegible]"
9010 FOR A=SGN PI TO LEN A$
9020 LET X=CODE A$(A)
9030 GOSUB 9070
9040 LET U=U+8
9050 NEXT A
9060 RETURN
9070 FOR I=V TO V-7 STEP -SGN PI
9080 LET X$=CHR$ (PEEK (7680+8*X
+V-I))
9090 FOR J=SGN PI TO 8
9100 IF X$<Z$(J) THEN GOTO 9130
9110 PLOT U+J-SGN PI,I
9120 LET X$=CHR$ (CODE X$-CODE Z
$(J))
9130 NEXT J
9140 NEXT I
9150 RETURN
```

1256

PROGRAMA ZX SPECTRUM

```
 100 INPUT "HORA (HHMM) :";A$
 110 PAPER 6: BORDER 6: CLS
 115 LET U=12
 120 LET V=26
 130 GO SUB 9000
 140 PAUSE 2640
 150 LET I=1+VAL A$(3 TO )
 155 LET J=VAL A$( TO 2)
 160 IF I=60 THEN GO SUB 200
 170 LET A$="0"+STR$ J
 173 LET A$=A$(LEN A$-1 TO )+"0"
+STR$ I
 175 LET A$=A$( TO 2)+A$(LEN A$-
1 TO )
 180 GO TO 110
 200 LET I=0
 210 LET J=J+1
 220 IF J=24 THEN LET J=0
 230 RETURN
9000 FOR K=0 TO 3
9010 LET X=15360+8*CODE A$(K+1)
9020 FOR I=0 TO 7
9030 LET Y=PEEK (X+I)
9040 FOR J=0 TO 7
9070 IF Y/2<>INT (Y/2) THEN PRIN
T AT 7+I,8*(K+1)-J-2*(K<2); INK
1+(K<2);"█"
9110 LET Y=INT (Y/2)
9120 NEXT J
9130 NEXT I
9140 NEXT K
9150 RETURN
```

1425

Damas

REFERENCIAS:

Modalidad: SLOW o FAST.
Dificultad de programación: muy difícil.
Categoría del juego: dos jugadores.
Dificultad del juego: difícil.

EL JUEGO:

Es el clásico juego de las damas, pero algo incompleto (1 K obliga) ya que no prevé la posibilidad de hacer damas. La regla del juego continúa siendo la clásica, pero los controles han sido un poco simplificados. Los movimientos de las piezas se introducen en una sola maniobra (posición de salida y dirección de desplazamiento), y las capturas se realizan automáticamente.

En cambio, ya que por cada jugada se realiza el cambio de jugador, será necesario pulsar Ø para volver a jugar en el caso de que hubiera un salto o si la jugada no fuera posible.

EL PROGRAMA:

El bucle I (2Ø-4Ø) dibuja el tablero (10 × 10 casillas) y los peones (X o 0).

La línea 5Ø indica quién debe jugar.

El cambio de jugador se trata por la línea 75 (introducida por 0).

Las líneas 8Ø y 9Ø calculan el desplazamiento, que se realizará, cuando sea posible, por el bucle I (12Ø-21Ø).

La línea 13Ø calcula la casilla de llegada (se tiene en cuenta, por si acaso se llegara a la línea Ø, en la línea 14Ø).

Las líneas 15Ø y 16Ø comprueban las casillas que hay que recorrer. La línea 18Ø trata el caso de un desplazamiento, y la 19Ø el de un salto. La línea 22Ø efectúa las eventuales modificaciones, y la línea 229 realiza el cambio de jugador.

PARA JUGAR:

En cuanto se pone en marcha el programa por RUN, se visualiza el tablero con sus peones y un carácter 0 invertido aparece por debajo, para indicar que el turno pasa al jugador que ha escogido 0.

Se juega indicando, por un número de 3 cifras (puede empezar por Ø), la línea y la columna de la pieza que tiene que desplazarse, y después la dirección del desplazamiento (codificado de 1 a 4), siguiendo la regla convencional:

```
4     1

3     2
```

Por ejemplo, 931 significa que el 0 situado en la línea 9, columna 3, se desplaza en la dirección 1 (no olvidar pulsar NEWLINE).

Del mismo modo, Ø63 significará pieza X, línea Ø, columna 6, dirección 3.

El cambio de jugador se efectúa automáticamente; será necesario pulsar Ø para volver a jugar, si la jugada ha sido rechazada o si ha habido captura (terminar siempre con NEWLINE). El programa no controla hacer damas y no prohíbe los desplazamientos hacia atrás.

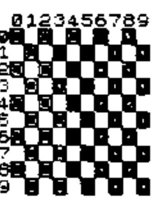
POSIBLES EXTENSIONES: Será bastante fácil empezar mejorando los controles (cambio de jugador, jugada hacia atrás que no corresponde a una captura, salida del tablero).

Se considerará el hacer damas, su desplazamiento y, llegado el caso, escribir un programa que permita al ZX mezclarse con éxito en apasionantes partidas.

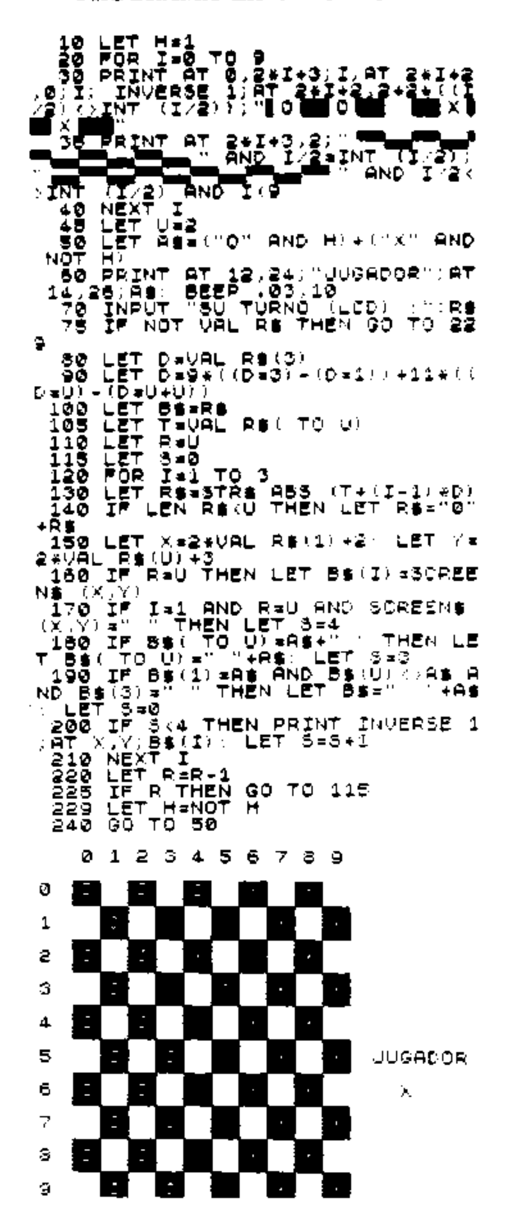
DAMAS

PROGRAMA ZX 81

```
  10 LET H=SGN PI
  20 FOR I=NOT PI TO PI*PI
  30 PRINT AT NOT PI,I+H;I;AT I+
H,NOT PI;I;TAB H+((I/(H+H))<>INT
 (I/(H+H)));"■ ■ ■ ■ ■"
  40 NEXT I
  45 LET U=H+H
  50 LET A$=("■" AND H)+("■" AND
 NOT H)
  60 PRINT AT CODE ")",U;A$
  70 INPUT R$
  75 IF NOT VAL R$ THEN GOTO COD
E " FAST "
  80 LET D=VAL R$(INT PI)
  90 LET D=INT (PI*PI)*((D=INT P
I)-(D=SGN PI))+INT (PI*PI+U)*((D
=U)-(D=U+U))
 100 LET B$=R$
 105 LET T=VAL R$( TO U)
 110 LET R=U
 120 FOR I=SGN PI TO INT PI
 130 LET R$=STR$ ABS (T+(I-SGN P
I)*D)
 140 IF LEN R$<U THEN LET R$="0"
+R$
 150 PRINT AT VAL R$(SGN PI)+SGN
 PI,VAL R$(U)+SGN PI;
 160 IF R=U THEN LET B$(I)=CHR$
PEEK (PEEK 16398+256*PEEK 16399)
 180 IF B$( TO U)=A$+"■" THEN LE
T B$( TO U)="■"+A$
 190 IF B$(SGN PI)=A$ AND B$(U)<
>A$ AND B$(INT PI)="■" THEN LET
B$="■■"+A$
 200 PRINT B$(I)
 210 NEXT I
 220 LET R=R-SGN PI
 225 IF R THEN GOTO 120
 229 LET H=NOT H
 240 GOTO CODE "M"
```



PROGRAMA ZX SPECTRUM



```
  10 LET H=1
  20 FOR I=0 TO 9
  30 PRINT AT 0,2*I+3;I,AT 2*I+2
,0;I; INVERSE 1;AT 2*I+2,2+2*((I
/2)<>INT (I/2));"■O■■O■■ ■■X■
■ X■■"
  35 PRINT AT 2*I+3,2;"▀▀▀▀▀▀▀▀
▀▀▀▀▀▀▀▀▀▀" AND I/2=INT (I/2);
"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀" AND I/2<
>INT (I/2) AND I<9
  40 NEXT I
  45 LET U=2
  50 LET A$=("O" AND H)+("X" AND
 NOT H)
  60 PRINT AT 12,24;"JUGADOR";AT
 14,28;A$: BEEP .03,10
  70 INPUT "SU TURNO (LCD) :";R$
  75 IF NOT VAL R$ THEN GO TO 22
9
  80 LET D=VAL R$(3)
  90 LET D=9*((D=3)-(D=1))+11*((
D=U)-(D=U+U))
 100 LET B$=R$
 105 LET T=VAL R$( TO U)
 110 LET R=U
 115 LET S=0
 120 FOR I=1 TO 3
 130 LET R$=STR$ ABS (T+(I-1)*D)
 140 IF LEN R$<U THEN LET R$="0"
+R$
 150 LET X=2*VAL R$(1)+2: LET Y=
2*VAL R$(U)+3
 160 IF R=U THEN LET B$(I)=SCREE
N$ (X,Y)
 170 IF I=1 AND R=U AND SCREEN$
(X,Y)=" " THEN LET S=4
 180 IF B$( TO U)=A$+" " THEN LE
T B$( TO U)=" "+A$: LET S=3
 190 IF B$(1)=A$ AND B$(U)<>A$ A
ND B$(3)=" " THEN LET B$="  "+A$
: LET S=0
 200 IF S<4 THEN PRINT INVERSE 1
;AT X,Y;B$(I): LET S=S+I
 210 NEXT I
 220 LET R=R-1
 225 IF R THEN GO TO 115
 229 LET H=NOT H
 240 GO TO 50
```

Anagramas

REFERENCIAS: Modalidad: SLOW o FAST (impuesto por el programa).
Dificultad de programación: difícil.
Categoría del juego: juego de reflejos (dos jugadores).
Dificultad del juego: según el nivel.

EL JUEGO: El ZX escoge letras al azar; el número de letras está en función del nivel de dificultad escogido. Después se mezclan las letras de la palabra obtenida de esta manera y se visualizan en la parte inferior de la pantalla; entonces se confecciona una nueva palabra de forma aleatoria, a partir de las mismas letras, y se visualiza debajo de la primera, y así sucesivamente; los dos jugadores (derecha e izquierda) disponen cada uno de una tecla (1 o Ø) y deben pulsarla en cuanto vean que se visualiza abajo de la pantalla una palabra que ya está presente en la pantalla; entonces se calcula su puntuación, y se muestra un mensaje correspondiente al resultado.

EL PROGRAMA: El nivel del juego se introduce en la línea 2Ø. El bucle I (3Ø-5Ø) realiza la elección de las letras.

Un segundo bucle I (8Ø-135) efectúa la mezcla, y la línea 1ØØ permite detectar la presión sobre una tecla.

La línea 14Ø visualiza la nueva palabra, y la línea 155 hace subir una línea en la pantalla las palabras que ya se han obtenido.

La línea 18Ø detecta al jugador que ha pulsado una tecla.

El bucle I (19Ø-26Ø) explora las palabras que están presentes en la pantalla, línea por línea, mientras que el bucle J (21Ø-24Ø) capta uno a uno los caracteres de estas palabras. Se obtiene, de esta manera, una nueva palabra que se compara con la palabra inicial en la línea 25Ø, y si son exactas aparece el resultado BRAVO.

Si no son exactas, se termina la partida con la visualización HA PERDIDO (línea 27Ø).

PARA JUGAR: Se pone en marcha el programa por RUN.

Después se introducirá el nivel de dificultad (en principio de 1 a 3) y se pulsará NEWLINE.

Las palabras se visualizan saliendo por la parte inferior de la pantalla y van subiendo a medida que se obtienen; si uno de los jugadores cree reconocer en la palabra que aparece una de las palabras que ya se han visualizado, deberá pulsar inmediatamente su tecla (1 o Ø), y se visualizará su resultado después de un corto instante. Entonces se termina la partida.

POSIBLES EXTENSIONES: Podrán utilizarse otras visualizaciones además de palabras, modificar los tiempos de visualización, participar más de dos jugadores, introducir sonido y color, etc.

ANAGRAMAS

PROGRAMA ZX 81

```
  10 PRINT "NIVEL (1/3)"
  20 INPUT N
  25 LET S$=""
  30 FOR I=SGN PI TO N+INT PI
  40 LET S$=S$+CHR$ INT (38+26*R
ND)
  50 NEXT I
  60 LET A$=""
  70 LET B$=S$
  80 FOR I=SGN PI TO LEN S$
  90 LET N=INT (LEN S$*RND+SGN P
I)
 100 IF INKEY$="0" OR INKEY$="1"
 THEN GO TO 170
 110 IF B$(N)="-" THEN GO TO 90
 120 LET A$=A$+B$(N)
 130 LET B$(N)="-"
 135 NEXT I
 140 PRINT AT 20,NOT PI;A$
 150 LET S$=A$
 155 SCROLL
 160 GO TO 60
 170 FAST
 180 LET H$=("DERECHA" AND INKEY
$="0")+("IZQUIERDA" AND INKEY$="
1")
 190 FOR I=SGN PI TO 18
 200 LET B$=""
 210 FOR J=SGN PI TO LEN S$
 220 PRINT AT I,J-SGN PI;
 230 LET B$=B$+CHR$ PEEK (PEEK 1
6398+256*PEEK 16399)
 240 NEXT J
 250 IF B$=S$ THEN PRINT TAB LEN
 S$+INT PI;"BRAVO ";H$;W
 260 NEXT I
 270 PRINT TAB LEN S$+INT PI;"HA
 PERDIDO ";H$
```

```
VQEI
EQVI
VQEI
EQIV
EVQI
VIEQ
VIQE      BRAVO DERECHA
IQVE
VQEI
EIQV
EIQV
IEQV
VIQE
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "NIVEL (1/3)"
  20 INPUT N
  25 LET S$=""
  30 FOR I=1 TO N+3
  40 LET S$=S$+CHR$ INT (65+25*R
ND)
  50 NEXT I
  60 LET A$=""
  70 LET B$=S$
  80 FOR I=1 TO LEN S$
  90 LET N=INT (LEN S$*RND+1)
 100 IF INKEY$="0" OR INKEY$="1"
 THEN GO TO 170
 110 IF B$(N)="-" THEN GO TO 90
 120 LET A$=A$+B$(N)
 130 LET B$(N)="-"
 135 NEXT I
 140 PRINT INVERSE 1; INK INT (5
*RND+1); BRIGHT 1;AT 20,0;A$
 145 PAUSE 8*LEN A$
 150 LET S$=A$
 155 POKE 23692,0: PRINT CHR$ (1
3)
 160 GO TO 60
 180 LET H$=("DERECHA" AND INKEY
$="0")+("IZQUIERDA" AND INKEY$="
1")
 190 FOR I=1 TO 18
 200 LET B$=""
 210 FOR J=1 TO LEN S$
 220 LET R$=SCREEN$ (I,J-1)
 230 LET B$=B$+R$
 240 NEXT J
 250 IF B$=S$ THEN PRINT FLASH 1
; INK 2;AT I,LEN S$+3;"BRAVO ";H
$: BEEP .3,10: BEEP .3,18: BEEP
.3,9: STOP
 260 NEXT I
 270 PRINT TAB LEN S$+3; INVERSE
 1;"HA PERDIDO ";H$: BEEP .8,-8:
 BEEP .3,-5: BEEP .5,-10
```



```
BRAVO DERECHA
```

Huevos

REFERENCIAS: Modalidad: FAST o SLOW.

Dificultad de programación: difícil.

Categoría del juego: solitario.

Dificultad del juego: mediana.

EL JUEGO: Un pájaro misterioso se desplaza al azar aleteando en la pantalla. Pone huevos (representados por 0) que el jugador (un + invertido) intentará recuperar. La puntuación (número de huevos recuperados) estará permanentemente visualizada, y la partida se desarrolla en un espacio de tiempo limitado.

EL PROGRAMA: El pájaro se dibuja en dos partes (para el movimiento) en las líneas 5Ø y 7Ø; su desplazamiento aleatorio se gestiona por el subprograma 2ØØ-27Ø (en particular, las líneas 23Ø a 26Ø).

La línea 1ØØ permite volver a cerrar el programa con la línea 5Ø.

El desplazamiento del jugador está dirigido por las líneas 16Ø y 17Ø, y la comprobación que permite detectar la presencia de un huevo se efectúa en las líneas 175, 178, 181.

La línea 186 permite la visualización de la puntuación.

El final del juego se detecta en la línea 214.

PARA JUGAR: Se pone en marcha el programa con RUN.

Entonces, el pájaro empieza a desplazarse al azar sobre la pantalla y a poner huevos.

Cuando el símbolo que representa al jugador (un + invertido) aparece en la pantalla, éste puede desplazarse utilizando las 4 teclas de desplazamiento. Desde entonces se intentará recuperar el máximo de huevos antes del final de la partida.

Se podrá parar el juego en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Sobre el mismo principio, podrán fácilmente imaginarse otros temas de juego. También puede introducirse color, sonidos, mejorar el grafismo y, eventualmente, definir un juego para varios jugadores, cada uno intentando recuperar el máximo de huevos.

HUEVOS

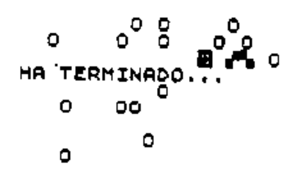
PROGRAMA ZX 81

```
  10 CLS
  15 LET P=CODE "█"
  20 LET X=CODE "£"
  30 LET Y=X
  40 LET R=NOT PI
  45 LET S=R
  47 LET T=R
  50 PRINT AT X,Y;"▀▀"
  57 IF INKEY$<>"" THEN GOSUB 15
0
  60 GOSUB CODE "COS "
  70 PRINT AT X,Y;"▄▀▄"
  90 GOSUB CODE "COS "
 100 GOTO CODE "M"
 150 PRINT AT R,S;" "
 160 LET R=R+(INKEY$="6" AND R<C
ODE "$")-(INKEY$="7" AND R>NOT P
I)
 170 LET S=S+(INKEY$="8" AND S<C
ODE "*")-(INKEY$="5" AND S>NOT P
I)
 175 PRINT AT R,S;
 178 LET A=PEEK 16398+256*PEEK 1
6399
 181 LET A=PEEK A=CODE "O"
 184 LET T=T+A
 186 PRINT AT CODE "(",CODE "£";
T
 187 RETURN
 200 LET A=X
 210 LET B=Y
 212 LET P=P-SGN PI
 214 IF NOT P THEN STOP
 215 PRINT AT R,S;"█"
 220 PAUSE CODE "PI"
 225 PRINT AT A,B;" O"
 230 LET X=ABS (X+2*RND-SGN PI)
 240 LET Y=ABS (Y+4*RND-SGN PI-S
GN PI)
 250 IF X>CODE "£" THEN LET X=SG
N PI
 260 IF Y>CODE "*" THEN LET Y=SG
N PI
 270 RETURN
```

24

PROGRAMA ZX SPECTRUM

```
  10 PAPER 6: INK 0: BORDER 1: C
LS
  15 LET P=200
  20 LET X=INT (RND*20)
  30 LET Y=INT (RND*20)
  40 LET R=0
  45 LET S=R
  47 LET T=R
  50 PRINT INK 5;AT X,Y;"▀▀"
  57 IF INKEY$<>"" THEN GO SUB 1
50
  60 GO SUB 200
  70 PRINT INK 5;AT X,Y;"▄▀▄"
  90 GO SUB 200
 100 GO TO 50
 150 PRINT AT R,S;" "
 160 LET R=R+(INKEY$="6" AND R<2
0)-(INKEY$="7" AND R>0)
 170 LET S=S+(INKEY$="8" AND S<3
0)-(INKEY$="5" AND S>0)
 178 LET A$=SCREEN$ (R,S)
 181 LET A=A$="O"
 184 LET T=T+A
 186 PRINT AT 21,12;"MARCADOR :"
;T
 187 RETURN
 200 LET A=X
 210 LET B=Y
 212 LET P=P-1: BEEP 0.1,10
 214 IF NOT P THEN BEEP .3,10: B
EEP .3,10: BEEP .3,8: PRINT FLAS
H 1;"HA TERMINADO...": STOP
 215 PRINT PAPER 2;AT R,S;"█"
 220 PAUSE 20
 225 PRINT AT A,B;" "; PAPER 4;"
O"
 230 LET X=ABS (X+2*RND-1)
 240 LET Y=ABS (Y+4*RND-2)
 250 IF X>19 THEN LET X=1
 260 IF Y>29 THEN LET Y=1
 270 RETURN
```



MARCADOR :40

Visualización giratoria (vertical)

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: anuncio publicitario.
Dificultad del juego: fácil.

EL JUEGO: El programa propuesto realiza el desfile de un texto, visualizado en caracteres gruesos, en la pantalla.

La utilización del generador de caracteres permite simplificar considerablemente el programa, lo mismo que la utilización de la instrucción SCROLL, propia del ZX, que garantiza el desfile.

EL PROGRAMA: La línea 2Ø permite la introducción del texto, que será ordenado en una variable alfanumérica A$.

El bucle A (3Ø-7Ø) realiza la visualización letra por letra, recurriendo al subprograma 2ØØ-31Ø que utiliza el generador de caracteres (línea 21Ø) con una técnica de impresión muy simplificada (bucle T (23Ø-28Ø), para asegurar la impresión de cada línea).

Los diferentes SCROLL (6Ø, 8Ø, 85, 29Ø) permiten el desfile vertical.

PARA JUGAR: Primeramente se pondrá en marcha el programa por la instrucción RUN; después se introduce el texto, y a continuación se pulsa NEWLINE y se mira. El programa es un bucle en sí mismo, y el desfile sólo podrá interrumpirse pulsando la tecla BREAK.

POSIBLES EXTENSIONES: Bastará con tener algo de imaginación (y disponer preferentemente de una capacidad de memoria mayor) para adaptar este programa a cualquier tipo de visualización publicitaria (se trata entonces de atraer la atención, por una parte, y de informar, por otra). Para este fin, podrá utilizarse color, personalizar los mensajes, utilizar varias líneas, etc.

VISUALIZACIÓN GIRATORIA (VERTICAL)

PROGRAMA ZX 81

```
 10 CLS
 20 INPUT A$
 30 FOR A=SGN PI TO LEN A$
 40 LET X=CODE A$(A)
 50 GO SUB 200
 60 SCROLL
 70 NEXT A
 80 SCROLL
 85 SCROLL
 90 GO TO 30
200 FOR Y=NOT PI TO 8
210 LET R=PEEK (7680+5*X+Y)
220 LET S=128
230 FOR T=15 TO 22
240 LET Z=INT (R/S)
250 LET R=R-Z*S
260 LET S=S/2
270 IF Z THEN PRINT AT 20,T;"█"
280 NEXT T
290 SCOLL
300 NEXT Y
310 RETURN
```





PROGRAMA ZX SPECTRUM

```
 10 CLS
 20 INPUT A$
 30 FOR A=1 TO LEN A$
 40 LET X=CODE A$(A)
 50 GO SUB 200
 60 PRINT AT 21,0: POKE 23692,0
: PRINT CHR$ 13
 70 NEXT A
 80 PRINT AT 20,0: POKE 23692,0
: PRINT CHR$ 13
 90 GO TO 30
200 FOR Y=0 TO 8
205 IF Y>=1 AND Y<=6 THEN INK Y
210 LET R=PEEK (PEEK 23606+256*
PEEK 23607+8*X+Y)
220 LET S=128
230 FOR T=15 TO 22
240 LET Z=INT (R/S)
250 LET R=R-Z*S
260 LET S=S/2
270 IF Z THEN PRINT AT 20,T;"█"
280 NEXT T
290 PRINT AT 20,0: POKE 23692,0
: PRINT CHR$ 13
300 NEXT Y
310 RETURN
```





De cuatro en cuatro

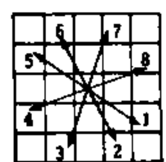
REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: reflexión (uno o más jugadores).
Dificultad del juego: difícil.

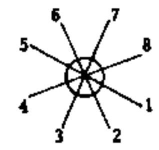
EL JUEGO: Puede practicarse por uno o dos jugadores.
En los dos casos, el objetivo consiste en cambiar los peones X por los peones 0; si solamente juega una persona, este cambio deberá realizarse en el menor número de jugadas posible. Los peones se desplazan alternativamente, respetando las reglas de desplazamiento del caballo en el ajedrez; no existe captura, y no puede saltarse un turno de jugada.

EL PROGRAMA: La variable alfanumérica A$, en la línea 15, contiene la tabla de posibles jugadas (desplazamiento del caballo).
Los bucles intercalados I (25-5Ø) y J (3Ø-47) dibujan el campo de juego y las posiciones de salida.
La línea 66 visualiza 0 o X, según el jugador que tenga el turno.
La línea 75 permite la visualización parpadeante del cursor (aquí un +), que indica en qué casilla se encuentra el jugador.
Las líneas 9Ø y 95 permiten el desplazamiento del cursor.
La línea 1ØØ comprueba si se ha pulsado sobre una tecla del 1 al 8, dando la dirección del desplazamiento.
La línea 18Ø vuelve a imprimir la figura que estaba debajo del cursor. La línea 187 transforma en número el símbolo de cifra pulsado.
La línea 19Ø efectúa el control de salida.
Las líneas 195 y 2ØØ efectúan el desplazamiento.
La línea 21Ø realiza el control de llegada.
El resultado se imprime por la línea 215; el cambio de jugador se realiza por la línea 22Ø y se imprime un . en el lugar de la salida (línea 223). El subprograma 229-24Ø se utiliza para los distintos controles.

PARA JUGAR: Se pone en marcha el programa por RUN. Se visualiza el campo de juego, igual que un símbolo que permite identificar al jugador que tiene el turno. El jugador puede escoger el peón que desea desplazar, manejando el cursor parpadeante gracias a las teclas D (→), I (←), A (↑), B (↓).

Para desplazar el peón, bastará con pulsar la dirección de desplazamiento, mientras el cursor esté encima de este peón, siguiendo el código:





Se rechazará la jugada si la posición de salida no es correcta o si la posición de llegada está ocupada. En caso contrario, se realiza la jugada y el turno pasa al otro jugador, el cual desplaza el cursor parpadeante hasta colocarlo sobre uno de los peones y realiza la jugada de la misma manera.

El jugador que gana es el que consigue situar primero todos sus peones en las posiciones iniciales ocupadas por el adversario.
Puede interrumpirse la partida en cualquier momento pulsando BREAK.
Si la maniobra de la tecla INKEY$ lleva en algún momento a salir del programa, deberá pulsarse GOTO 66 para continuar.

POSIBLES EXTENSIONES: Puede utilizarse el color, mejorar los controles, prever una comprobación del final, utilizar un campo de juego distinto; puede también considerarse la escritura de un programa que permita al ZX participar en el juego con muchas posibilidades de ganarlo.

DE CUATRO EN CUATRO

PROGRAMA ZX 81

```
  10 LET H=SGN PI
  15 LET A$="5665322356"
  18 LET J=VAL INKEY$
  25 FOR I=H TO CODE "▌"
  30 FOR J=H TO CODE "▌"
  35 PRINT AT I+I,J+J;"."
  40 IF I<INT PI AND J<INT PI TH
EN PRINT AT I+I,J+J;"O"
  45 IF I>INT PI AND J>INT PI TH
EN PRINT AT I+I,J+J;"X"
  47 NEXT J
  50 NEXT I
  55 LET X=H
  60 LET Y=X
  66 PRINT AT CODE "▪",NOT PI;"O
" AND H;"X" AND NOT H
  70 GOSUB CODE " FAST "
  72 LET I=J
  75 PRINT "+"
  80 LET A=X
  85 LET B=Y
  90 LET X=X+(INKEY$="B" AND X<C
ODE "▌")-(INKEY$="H" AND X>SGN P
I)
  95 LET Y=Y+(INKEY$="D" AND Y<C
ODE "▌")-(INKEY$="G" AND Y>SGN P
I)
 100 IF INKEY$<"9" AND INKEY$>"0
" THEN GOTO CODE "█"
 160 PRINT AT A+A,B+B;CHR$ I
 165 GOTO CODE "PI"
 167 LET J=VAL STR$ CODE INKEY$
 190 IF J>=PI*PI OR (I<>CODE "O"
 AND H) OR (I<>CODE "X" AND NOT
H) THEN GOTO CODE "█"
 195 LET X=X+VAL A$(J)-CODE "▪"
 200 LET Y=Y+VAL A$(J+2)-CODE "▪
"
 205 GOSUB CODE " FAST "
 210 IF J<>CODE "." THEN GOTO CO
DE "█"
 215 PRINT "O" AND H;"X" AND NOT
 H
 220 LET H=ABS (H-SGN PI)
 223 LET I=CODE "."
 225 GOTO CODE "█"
 229 PRINT AT X+X,Y+Y;
 235 LET J=PEEK (PEEK 16398+256*
PEEK 16399)
 240 RETURN
```

```
. O . . .
O O . . .
. O . . .
. . . X X
. . . X X

X
```

PROGRAMA ZX SPECTRUM

```
  10 LET H=1
  15 LET A$="5665322356"
  25 FOR I=H TO 5
  30 FOR J=H TO 5
  35 PRINT AT 0,I+I;I;AT I+I,0;I
;AT I+I,J+J;"."
  40 IF I<3 AND J<3 THEN PRINT A
T I+I,J+J;"O"
  45 IF I>3 AND J>3 THEN PRINT A
T I+I,J+J;"X"
  47 NEXT J
  50 NEXT I
  55 LET X=H
  60 LET Y=X
  66 PRINT AT 13,0;"JUGADOR ";"O
" AND H;"X" AND NOT H
  70 GO SUB 229
  72 LET I=J
  75 PRINT "+"
  80 LET A=X
  85 LET B=Y
  90 LET X=X+(INKEY$="B" AND X<5
)-(INKEY$="H" AND X>1)
  95 LET Y=Y+(INKEY$="D" AND Y<5
)-(INKEY$="G" AND Y>1)
 100 IF INKEY$<"9" AND INKEY$>"0
" THEN GO TO 167
 160 PRINT AT A+A,B+B;CHR$ I
 165 GO TO 66
 167 LET J=VAL INKEY$
 190 IF J>9 OR (I<>CODE "O" AND
H) OR (I<>CODE "X" AND NOT H) TH
EN GO TO 160
 195 LET X=X+VAL A$(J)-4
 200 LET Y=Y+VAL A$(J+2)-4
 205 GO SUB 229
 210 IF J<>CODE "." THEN GO TO 1
60
 215 PRINT "O" AND H;"X" AND NOT
 H
 220 LET H=ABS (H-1)
 223 LET I=CODE "."
 225 GO TO 160
 229 PRINT AT X+X,Y+Y;
 235 LET J=CODE SCREEN$ (X+X,Y+Y
)
 240 RETURN
```

```
    1 2 3 4 5
  1 O O . . .
  2 O . . . .
  3 . X . . .
  4 . . O . X
  5 . . . X X

  JUGADOR O
```

Revoltijo de letras

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: difícil.
Categoría del juego: de reflexión.
Dificultad del juego: difícil.

EL JUEGO: ¿Quién no ha tenido entre sus manos, un día u otro, uno de estos pequeños rompecabezas en los que se intenta desplazar de una posición a otra pequeños cuadrados de plástico de colores, hasta la obtención de un resultado preciso? En su tiempo, el 14-15 de SAM LOYD tuvo tanto éxito como el actual RUBIK'S.

El rompecabezas que aquí se propone utiliza 35 letras, dispuestas al azar en un cuadrado de 6 × 6 casillas, y estas letras, colocadas de nuevo en su orden inicial, pueden permitir la reconstrucción de un proverbio muy conocido.

Una casilla gris permite desplazar las letras, y un contador visualiza permanentemente el número de intentos: ¿llegará a encontrarse el proverbio?

EL PROGRAMA: La frase que hay que reconstruir (que podrá, por supuesto, modificarse) se sitúa en una variable B$, en la línea 25.

La mezcla de las letras y la impresión se realiza por el bucle I (4Ø-12Ø).

La línea 17Ø realiza la visualización del cuadrado gris; éste se desplaza por las líneas 21Ø y 22Ø, utilizando las 4 flechas.

Las líneas 24Ø y 25Ø vuelven a imprimir la letra referida en el lugar del cuadrado gris.

La línea 26Ø contabiliza el número de intentos, que se imprime gracias a la línea 16Ø.

PARA JUGAR: Se pone en marcha el programa por RUN.

La partida será, evidentemente, más difícil si no se conoce el proverbio que tiene que reconstruirse. Pueden desplazarse las letras y hacerlas pasar al lugar del cuadrado gris con la ayuda de las 4 teclas de dirección (5, 6, 7, 8).

El número de intentos estará permanentemente visualizado a la derecha del rompecabezas.

Puede pararse en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: El mismo principio puede aplicarse a todo tipo de juegos de revoltijo (manejo de números, de colores y, eventualmente, de sonidos). Pueden también preverse varias frases, con una elección efectuada por un generador aleatorio, etc.

REVOLTIJO DE LETRAS

PROGRAMA ZX 81

```
   5 LET A$=""
  25 LET R$="SIEMPRENECESITASAUN
OMASPEQUENOQUETU"
  35 RAND
  40 FOR I=SGN PI TO 35
  50 LET X=INT (RND*35+1)
  60 IF R$(X)="-" THEN GOTO 50
  70 LET A$=A$+R$(X)
  80 LET R$(X)="-"
 110 PRINT AT INT (I/6),I-6*INT
(I/6);A$(I)
 120 NEXT I
 130 LET X=NOT PI
 140 LET Y=X
 150 LET S=X
 160 PRINT AT NOT PI,12;S
 170 PRINT AT X,Y;"█"
 180 PAUSE 4E4
 190 LET I=X
 200 LET J=Y
 210 LET X=X+(INKEY$="6" AND X<5
)-(INKEY$="7" AND X>NOT PI)
 220 LET Y=Y+(INKEY$="8" AND Y<5
)-(INKEY$="5" AND Y>NOT PI)
 230 PRINT AT X,Y;
 240 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
 250 PRINT AT I,J;CHR$ C
 260 LET S=S+SGN PI
 270 GOTO 160
```

```
OURPNI        7
OETEAM
ESO█IE
MNOUNS
SISECT
AEAUEP
```

PROGRAMA ZX SPECTRUM

```
   5 LET A$=""
  10 BORDER 6: PAPER 7: CLS
  25 LET R$="151I2E3M3P3R3E3N3E3
C4E4S4I4T4A4S5A5U3N3O1M1A1S1P2E2
Q2U2E2N3O3Q3U4E4T4U"
  35 RANDOMIZE
  40 FOR I=1 TO 35
  50 LET X=INT (RND*35+1)
  60 IF R$(2*X-1 TO 2*X)="--" TH
EN GO TO 50
  70 LET A$=A$+R$(2*X-1 TO 2*X)
  80 LET R$(2*X-1 TO 2*X)="--"
 110 PRINT INVERSE 1; BRIGHT 1;
INK VAL A$(2*I-1);AT 2+INT (I/6)
,2+I-6*INT (I/6);A$(2*I)
 120 NEXT I
 130 LET X=0
 140 LET Y=X
 150 LET S=X
 160 PRINT INVERSE 1;AT 0,9;"NUM
ERO DE INTENTOS :";S
 170 PRINT FLASH 1;AT X+2,Y+2;"█
"
 180 PAUSE 4E4
 190 LET I=X
 200 LET J=Y
 210 LET X=X+(INKEY$="6" AND X<5
)-(INKEY$="7" AND X>0)
 220 LET Y=Y+(INKEY$="8" AND Y<5
)-(INKEY$="5" AND Y>0)
 240 LET C$=SCREEN$ (X+2,Y+2): L
ET B=ATTR (X+2,Y+2)
 245 IF B>125 OR B<121 THEN LET
B=127: BEEP .3,18
 250 PRINT INVERSE 1; BRIGHT 1;
INK (B-120);AT I+2,J+2;C$: BEEP
.03,8
 260 LET S=S+1
 270 GO TO 160
```



Blancos

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: de azar.
Dificultad del juego: fácil.

EL JUEGO: Consiste simplemente en tirar al blanco, intentando obtener el máximo de puntos; el número de blancos (de 1 a 3) se escoge al principio de la partida, y la puntuación realizada se visualizará permanentemente. Si se toca el borde de un blanco, la partida se acaba. Cuanto más alto sea el número en el que se caiga, mayor será la puntuación obtenida.

EL PROGRAMA: El número de blancos se introduce en la línea 5.

El bucle I (3Ø-5Ø) efectúa la impresión de los blancos.

La línea 75 visualiza permanentemente la puntuación.

La bola se visualiza por la línea 8Ø.

El bucle I (1ØØ-14Ø) realiza la impresión de las cifras contenidas en los blancos y permite el desplazamiento de la bola (línea 124).

La línea 15Ø comprueba si se ha tirado, y la línea 16Ø si la bola aún está en la línea de salida. La línea 19Ø hace avanzar la bola una posición a la derecha, y la línea 2ØØ verifica si se han alcanzado o no los blancos; después las líneas 21Ø y 22Ø comprueban lo que se ha tocado.

La línea 24Ø aumenta la puntuación; si se desea que el número de puntos marcados corresponda a la cifra visualizada habrá que escribir: 24Ø LET S = S + VAL CHR$ C.

Las líneas 255 y 26Ø se ocupan del final de la partida.

PARA JUGAR: Poner en marcha el programa por RUN y teclear el número de blancos que se deseen, seguido de NEWLINE.

Los blancos se visualizan y la partida empieza. La bola, representada por un Ø, se visualiza en la parte inferior a la izquierda; se la puede desplazar con las flechas de arriba y de abajo (teclas 6 y 7), y puede tirarse en cualquier momento pulsando Ø.

La puntuación se modificará en función del resultado, y aparecerá una nueva bola.

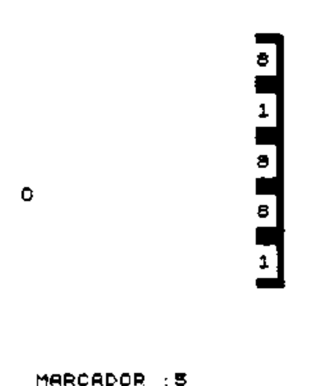
Es posible parar el juego en cualquier momento pulsando BREAK; si no, el juego se para al cabo de un cierto número de tiradas.

POSIBLES EXTENSIONES: Se podrá aumentar sin dificultad el número de blancos, modificar la regla de aparición de los números en los blancos para que no sea tan aventurada, disponer que la bola sea representada por una cifra que coincida a la llegada con la del blanco, etc.

BLANCOS

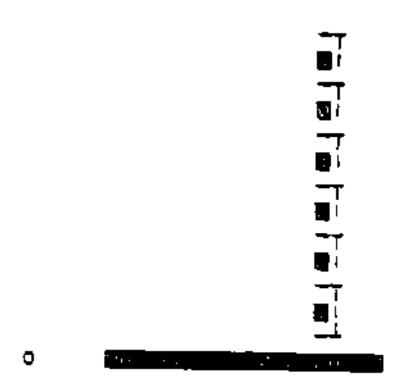
PROGRAMA ZX 81

```
   5 INPUT N
   7 LET P=CODE "▀"
   8 LET J=P+P
  10 LET S=NOT PI
  30 FOR I=SGN PI TO N
  40 PRINT AT INT PI*I-PI,J;" ▀▜
";AT INT PI*I-2,J;"  ▐";AT I*INT
 PI-SGN PI,J;" ▄▟"
  50 NEXT I
  60 LET K=NOT PI
  70 LET A=J
  75 PRINT AT A,N;"MARCADOR ";S
  80 PRINT AT A,K;"O"
  90 LET D=A
 100 FOR I=SGN PI TO N
 105 LET U=INT (P*RND)
 110 PRINT AT I*INT PI-2,J+SGN P
I;U
 124 LET A=A+(INKEY$="6")-(INKEY
$="7")
 140 NEXT I
 146 PRINT AT D,K;" ";
 150 IF INKEY$="0" THEN GOTO 190
 160 IF NOT K THEN GOTO 80
 190 LET K=K+SGN PI
 200 IF K<=J THEN GOTO 80
 210 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
 220 IF C=CODE "▄" OR C=CODE "▀"
 THEN PRINT "HA PERDIDO";W
 240 LET S=S+C
 255 LET P=P-SGN PI
 260 IF NOT P THEN STOP
 270 GOTO CODE "W"
```



PROGRAMA ZX SPECTRUM

```
   5 INPUT "NUMERO DE BLANCOS :"
;N
   7 LET P=10
   8 LET J=P+P
  10 LET S=0: LET B=1
  30 FOR I=1 TO N
  40 PRINT PAPER 4;AT 3*I-3,J;"
__";AT 3*I-2,J;"  |";AT 3*I-1,J;
"  |";AT 3*I,J;"  |"; OVER 1;AT
3*I,J;" __"
  50 NEXT I
  60 LET K=0
  70 LET A=J
  75 PRINT INVERSE 1; INK 1;AT A
,N;"MARCADOR :";S; INK 3;" BOLAS
 ";B
  80 PRINT AT A,K;"O"
  90 LET D=A
 100 FOR I=1 TO N
 105 LET U=INT (P*RND)
 110 PRINT INVERSE 1; INK 2;AT 3
*I-1,J+1;U
 124 LET A=A+(INKEY$="6")-(INKEY
$="7")
 140 NEXT I
 146 PRINT AT D,K;" ";
 150 IF INKEY$="0" THEN BEEP .03
,10: LET B=B+1: GO TO 190
 160 IF NOT K THEN GO TO 80
 190 LET K=K+1
 200 IF K<=J THEN GO TO 80
 210 LET C$=SCREEN$ (A,K)
 220 IF C$="_" OR C$="|" THEN PR
INT "HA PERDIDO";W
 240 LET S=S+VAL C$: BEEP .3,10:
 BEEP .3,18: BEEP .3,8
 255 LET P=P-1
 260 IF NOT P THEN STOP
 270 GO TO 60
```



Gotas

REFERENCIAS: Modalidad: SLOW o FAST.
Dificultad de programación: difícil.
Categoría del juego: dibujo.
Dificultad del juego: ninguna.

EL JUEGO: Es un verdadero programa de dibujo animado que, de hecho, puede considerarse como un interludio; se dibuja un recipiente en la pantalla, en forma de copa, el cual ya contiene algunos obstáculos. Una gota de agua, simbolizada por un cuadrado gris, aparece encima del recipiente y cae sobre éste; ésta y otras gotas van llenándolo a medida que caen. Cuando la gota encuentra un obstáculo, se determina su trayectoria completamente al azar: puede ir a la derecha o a la izquierda, hasta que ya no exista ninguna posibilidad. Entonces cae otra gota, y así sucesivamente hasta el desbordamiento final.

EL PROGRAMA: Los dos bucles intercalados I (1Ø-5Ø) y J (2Ø-4Ø) dibujan el recipiente y los obstáculos.

La línea 1ØØ dibuja la gota.

La línea 12Ø borra la gota anterior, y la línea 14Ø bifurca a un subprograma (228-23Ø) que permite determinar si se tropieza con el obstáculo o no.

Si no se encuentra, se va a la línea 212 y la gota continúa cayendo.

Si se encuentra, la gota va hacia la izquierda o hacia la derecha (línea 16Ø), según el valor definido por K, en la línea 9Ø.

De nuevo se mira si hay algún obstáculo o no, o si la gota puede continuar cayendo (líneas 17Ø, 18Ø, 19Ø).

Según el resultado de la comprobación, se utilizarán la 2ØØ, 212, 218 o 223.

PARA JUGAR: RUN pone en marcha el dibujo del recipiente y de los obstáculos; la gota aparece, empieza a caer y termina por encontrar un sitio. Otra gota aparece, y vuelve a empezarse.

El programa se para por sí mismo cuando el recipiente se desborda; también se puede parar en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Por supuesto, puede añadirse color y sonido, y también considerar un juego en el que un pequeño personaje debe escapar de las gotas e incluso de ahogarse.

GOTAS

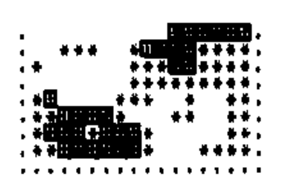
PROGRAMA ZX 81

```
  10 FOR I=CODE ":" TO CODE "*"
  20 FOR J=NOT PI TO CODE "("
  30 PRINT AT CODE "$",NOT PI;"█
";TAB CODE "(";"█";AT I,J;"█"
  35 PRINT AT I,SGN PI+(RND*CODE
 "$");" " AND I<>CODE "=" AND R
ND<.6
  40 NEXT J
  50 NEXT I
  60 LET Y=INT PI*INT PI
  70 LET X=Y
  90 LET K=2*(RND<.5)-SGN PI
  95 LET I=NOT PI
 100 PRINT AT X,Y;"█"
 110 PAUSE CODE "PI"
 120 PRINT AT X,Y;" "
 130 PRINT TAB Y;
 140 GOSUB CODE " SLOW "
 150 IF NOT J THEN GOTO CODE "US
R "
 160 PRINT AT X,Y+K;
 170 GOSUB CODE " SLOW "
 180 IF NOT J THEN GOTO CODE " A
ND "
 190 IF NOT I THEN GOTO CODE " T
O "
 200 PRINT AT X,Y;"█"
 210 GOTO CODE "W"
 212 LET X=X+SGN PI
 213 GOTO 90
 218 LET Y=Y+K
 219 LET I=SGN PI
 220 GOTO 100
 223 LET K=-K
 224 LET I=SGN PI
 225 GOTO CODE "█"
 228 LET J=PEEK 16398+256*PEEK 1
6399
 229 LET J=PEEK J=CODE "█" OR PE
EK J=CODE "█"
 230 RETURN
```



PROGRAMA ZX SPECTRUM

```
  10 FOR I=12 TO 20
  20 FOR J=1 TO 16
  30 PRINT PAPER 1;AT I,0;".";AT
 I,17;".";AT 20,J;"."; PAPER 4;A
T I,J;"*" AND I<>12 AND I<>20
  35 PRINT AT I,1+(RND*13);" "
AND I<>20 AND RND<.8
  40 NEXT J
  50 NEXT I
  60 LET Y=INT (7*RND+6)
  70 LET X=6
  90 LET K=2*(RND<.5)-1
  95 LET I=0
 100 PRINT INVERSE 1; INK 2;AT X
,Y;"#"
 110 PAUSE 10: BEEP .02,0
 120 PRINT AT X,Y;" "
 130 LET U=X+1: LET V=Y
 140 GO SUB 228
 150 IF NOT J THEN GO TO 212
 160 LET U=X: LET V=Y+K
 170 GO SUB 228
 180 IF NOT J THEN GO TO 218
 190 IF NOT I THEN GO TO 223
 200 PRINT INVERSE 1; INK 2;AT X
,Y;"#"
 210 GO TO 60
 212 LET X=X+1
 213 GO TO 90
 218 LET Y=Y+K
 219 LET I=1
 220 GO TO 100
 223 LET K=-K
 224 LET I=1
 225 GO TO 160
 228 LET C$=SCREEN$ (U,V)
 229 LET J=C$<>" "
 230 RETURN
```



Obstáculos

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: bastante difícil.
Categoría del juego: juego de reflejos (para un jugador).
Dificultad del juego: mediana.

EL JUEGO: Se desplazan obstáculos sobre la pantalla, que van al encuentro de un pequeño coche simbolizado por una Y invertida, que dispone de la posibilidad de desplazarse hacia la izquierda o hacia la derecha para evitarlos durante el mayor tiempo posible.

Se visualizará permanentemente la distancia recorrida, y el juego termina cuando el coche tropieza con el borde del campo de juego o con un obstáculo.

EL PROGRAMA: La línea 3Ø (instrucción SCROLL) hace avanzar los obstáculos hacia el coche del jugador, que se imprime gracias a la línea 4Ø.

Las líneas 6Ø, 7Ø y 8Ø confeccionan el recorrido (las líneas 7Ø y 8Ø, aunque idénticas, son las dos necesarias, ya que el valor de RND, en principio, no será el mismo en estas dos líneas).

La línea 9Ø visualiza el recorrido y la puntuación.

La línea 1ØØ permite el desplazamiento del vehículo.

Las líneas 11Ø, 12Ø y 13Ø permiten controlar la existencia de un obstáculo y paran el programa en caso necesario.

La línea 15Ø incrementa la puntuación.

PARA JUGAR: Se pone en marcha el programa pulsando RUN.

El recorrido y los obstáculos aparecen en la parte inferior de la pantalla y suben al encuentro del coche, representado por una Y invertida. El coche puede desplazarse con la ayuda de las flechas de desplazamiento horizontal (teclas 5 y 8).

El tropiezo con un obstáculo interrumpe el desarrollo del programa y provoca la visualización de la puntuación.

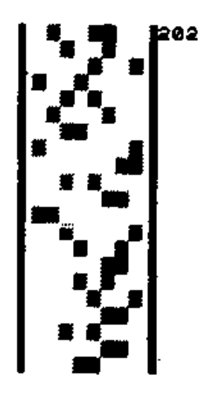
Sin embargo, también puede salirse del programa en cualquier momento utilizando la tecla BREAK.

POSIBLES EXTENSIONES: Será bastante fácil, sobre todo disponiendo de un ZX Spectrum, mejorar el dibujo del coche, del recorrido, de los obstáculos. También podrá introducirse color, admitir un cierto número de accidentes, grabar la mejor puntuación (para varios jugadores), dibujar eventualmente varios coches, etc.

OBSTÁCULOS

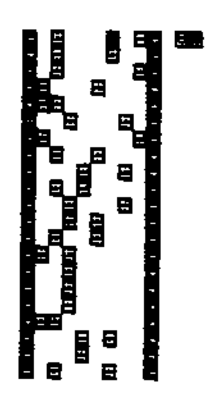
PROGRAMA ZX 81

```
  10 LET Y=5
  20 LET S=NOT PI
  30 SCROLL
  40 PRINT AT NOT PI,Y;"█"
  60 LET A$="▌          ▐"
  70 LET A$(7+RND+2)="▒"
  80 LET A$(7+RND+2)="▒"
  90 PRINT AT 20,NOT PI;A$;AT NO
T PI,PI+PI;S
 100 LET Y=Y+(INKEY$="8")-(INKEY
$="5")
 110 PRINT AT SGN PI,Y;
 120 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
 130 IF C=CODE "▒" OR C=CODE "▌"
 OR C=CODE "▐" THEN STOP
 150 LET S=S+SGN PI
 160 GOTO 30
```



PROGRAMA ZX SPECTRUM

```
  10 LET Y=5
  20 LET S=0
  30 PRINT AT 20,0: POKE 23692,0
: PRINT CHR$ 13
  40 PRINT INK 2;AT 0,Y;"█"
  60 LET A$="+[redacted]+"
  70 LET A$(7*RND+2)="#"
  80 LET A$(7*RND+2)="#"
  90 PRINT INK 1; INVERSE 1;AT 2
0,0;A$;AT 0,11;S
 100 LET Y=Y+(INKEY$="8")-(INKEY
$="5")
 120 LET C$=SCREEN$ (1,Y)
 130 IF C$="#" OR C$="+" THEN ST
OP
 150 LET S=S+1
 160 GO TO 30
```



Capturas en el espacio

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: reflejos (un solo jugador).
Dificultad del juego: mediana.

EL JUEGO: Una nave espacial navega en un peligroso campo de meteoritos, y su misión consiste en apoderarse de la mayor cantidad posible de mineral de uranio, un metal indispensable que sólo puede encontrarse en estos lugares especialmente peligrosos.

Debe maniobrarse para recoger el uranio sin chocar con ningún meteorito.

EL PROGRAMA: La línea 2Ø permite que aparezcan los meteoritos.

La línea 3Ø hará aparecer el mineral de uranio.

Los desplazamientos en la pantalla se resuelven por las instrucciones SCROLL, en las líneas 3Ø y 4Ø.

Las líneas 45 a 48 controlan el desarrollo del juego (se ha chocado con un meteorito o se ha capturado un bloque de uranio).

El choque con un meteorito provoca el final de la partida.

La línea 6Ø imprime la nave espacial, y su desplazamiento se controla por la línea 7Ø.

PARA JUGAR: Se pone en marcha el programa con RUN, y los meteoritos empiezan a subir en la pantalla, lo mismo que los bloques de uranio.

Las teclas de desplazamiento horizontal (5 y 8) serán utilizadas para desplazar la nave y recuperar el uranio; pero el menor choque con un meteorito provocará el final de la partida y la visualización de la cantidad de uranio capturado.

También podrá pararse el programa en cualquier momento pulsando BREAK.

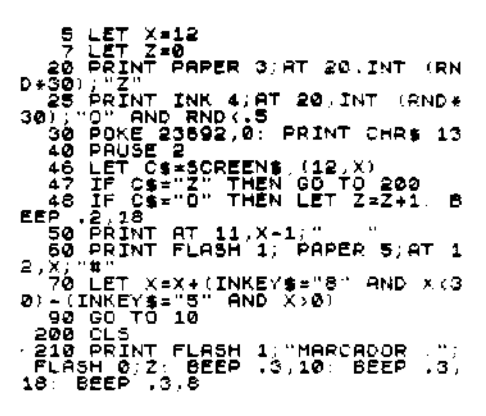
POSIBLES EXTENSIONES: Puede hacerse el juego cada vez más difícil, aumentando progresivamente el número de meteoritos, o también cronometrando el tiempo de duración de la partida.

También es posible prever una visualización permanente de la puntuación, introducir otra nave espacial maniobrada por una potencia extranjera que también quiere coger uranio, etc.

CAPTURAS EN EL ESPACIO

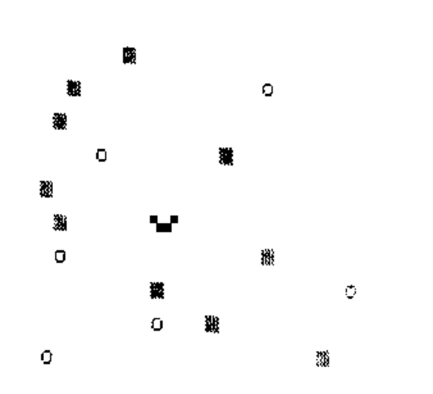
PROGRAMA ZX 81

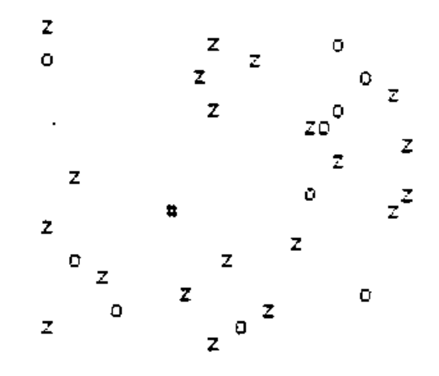
```
   5 LET X=12
   7 LET Z=NOT PI
  20 PRINT AT 20,INT (RND*30),"▒
"
  25 PRINT AT 20,INT (RND*30);"O
" AND RND<.5
  30 SCROLL
  40 SCROLL
  45 PRINT AT 12,X;
  46 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
  47 IF CHR$ C="▒" THEN GOTO 200
  48 IF CHR$ C="O" THEN LET Z=Z+
SGN PI
  50 PRINT AT 10,X-2;"         "
  60 PRINT AT 12,X;" "
  70 LET X=X+(INKEY$="8" AND X<3
0)-(INKEY$="5" AND X>NOT PI)
  90 GOTO 10
 200 CLS
 210 PRINT "MARCADOR ",Z
```



PROGRAMA ZX SPECTRUM

```
   5 LET X=12
   7 LET Z=0
  20 PRINT PAPER 3;AT 20,INT (RN
D*30);"Z"
  25 PRINT INK 4;AT 20,INT (RND*
30);"O" AND RND<.5
  30 POKE 23692,0: PRINT CHR$ 13
  40 PAUSE 2
  46 LET C$=SCREEN$ (12,X)
  47 IF C$="Z" THEN GO TO 200
  48 IF C$="O" THEN LET Z=Z+1: B
EEP .2,18
  50 PRINT AT 11,X-1;"   "
  60 PRINT FLASH 1; PAPER 5;AT 1
2,X;"#"
  70 LET X=X+(INKEY$="8" AND X<3
0)-(INKEY$="5" AND X>0)
  90 GO TO 10
 200 CLS
 210 PRINT FLASH 1;"MARCADOR ";
 FLASH 0;Z: BEEP .3,10: BEEP .3,
18: BEEP .3,8
```





Anuncio publicitario (2)

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: gráfico.
Dificultad del juego: ninguna.

EL JUEGO: Este programa permitirá la visualización giratoria de un texto en tamaño grande; utiliza las instrucciones SCROLL y el generador de caracteres, pero tiene la particularidad de hacer que los caracteres desfilen inclinados horizontalmente, lo que implica inclinar la pantalla de visualización, en el sentido de la altura, para obtener una visualización gigante con desfile horizontal, de la derecha a la izquierda.

EL PROGRAMA: La línea 2Ø permitirá la introducción del texto.

El bucle A (3Ø-7Ø) realiza la visualización letra a letra.

El subprograma, que va de la línea 2ØØ a la línea 32Ø, visualiza cada una de las letras en tamaño grande recurriendo al generador de caracteres (línea 21Ø) y realizando la decodificación de cada carácter verticalmente y no horizontalmente (bucle I, de 2ØØ a 22Ø).

Otro bucle I (25Ø-28Ø) realiza la visualización.

PARA JUGAR: Primero se pondrá en marcha el programa con RUN, y después se introducirá el texto a visualizar seguido de NEWLINE.

El texto empieza a desfilar en la pantalla de una forma curiosa, y deberá inclinarse el televisor por un lado para poder leerlo.

La anchura de los caracteres dibujados no permite que haya simultáneamente más de tres en la pantalla.

Puede pararse el programa en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Será posible volver a tener en cuenta esta idea (¡sin tener que inclinar necesariamente el televisor!), para realizar toda clase de anuncios de fantasía (colores, caracteres especiales, etc.).

También podrá modificarse la línea 6Ø sin tocar demasiado el programa, de tal manera que la visualización de una P en tamaño grande se realice con letras P, y lo mismo para las otras letras.

ANUNCIO PUBLICITARIO (2)

PROGRAMA ZX 81

```
  10 CLS
  15 DIM N(INT PI+INT PI)
  20 INPUT A$
  30 FOR A=SGN PI TO LEN A$
  40 LET X=CODE A$(A)
  50 GOSUB CODE "COS "
  60 SCROLL
  70 NEXT A
  80 SCROLL
  85 SCROLL
  90 GOTO CODE "2"
 200 FOR I=SGN PI TO INT PI+INT PI
 210 LET N(7-I)=PEEK (7680+8*X+I)
 220 NEXT I
 230 LET S=CODE "RND"
 240 LET Y=INT PI
 250 FOR I=SGN PI TO INT PI+INT PI
 260 IF N(I)-S>=NOT PI THEN PRINT AT CODE "=";Y;"█"
 265 LET N(I)=N(I)-(S AND N(I)-S>=NOT PI)
 270 LET Y=Y+SGN PI
 280 NEXT I
 290 LET S=S/2
 300 SCROLL
 310 IF S>SGN PI THEN GOTO 240
 320 RETURN
```



PROGRAMA ZX SPECTRUM

```
  10 PAPER 6: CLS
  15 DIM N(6)
  20 INPUT "SU TEXTO :";A$
  30 FOR A=1 TO LEN A$
  40 LET X=CODE A$(A)
  50 GO SUB 200
  60 PRINT AT 20,0: POKE 23692,0: PRINT CHR$ 13
  70 NEXT A
  80 PRINT AT 20,0: POKE 23692,0: PRINT CHR$ 13
  90 GO TO 30
 200 FOR I=1 TO 6
 210 LET N(7-I)=PEEK (15360+8*X+I)
 220 NEXT I
 230 LET S=64
 240 LET Y=3
 250 FOR I=1 TO 6
 260 IF N(I)-S>=0 THEN PRINT INK 2; BRIGHT 1;AT 20,Y;"█"
 265 LET N(I)=N(I)-(S AND N(I)-S>=0)
 270 LET Y=Y+1
 280 NEXT I
 290 LET S=S/2
 300 PRINT AT 20,0: POKE 23692,0: PRINT CHR$ 13
 310 IF S>1 THEN GO TO 240
 320 RETURN
```



Anomalía

REFERENCIAS:

Modalidad: FAST.

Dificultad de programación: difícil.

Categoría del juego: solitario.

Dificultad del juego: mediana.

EL JUEGO:

Se funda esencialmente en una buena capacidad de observación. Dos rejillas, de 25 casillas cada una, se dibujan en la pantalla. Símbolos idénticos se visualizan en diferentes sitios de las dos rejillas, aunque uno de los símbolos ha sido modificado en la rejilla de la parte derecha. El juego consiste en averiguar cuál es, introduciendo sus coordenadas en el orden línea-columna.

EL PROGRAMA:

El bucle I (2Ø-24) imprime las cifras de localización de las rejillas.

El bucle principal I (47-1ØØ) imprime los símbolos de las dos rejillas en un orden diferente. La línea 5Ø escoge el carácter que hay que imprimir.

El subprograma (1ØØØ-1Ø4Ø) se ocupa de que la impresión se haga en un sitio libre (se recurre a él dos veces para las dos rejillas).

Las líneas 12Ø y 13Ø cambian de forma aleatoria uno de los caracteres.

Las coordenadas del punto que se ha probado son introducidas en las líneas 14Ø y 16Ø, y el control se efectúa por las líneas 17Ø y 18Ø.

Las líneas 185 a 21Ø se ocupan de la visualización del resultado.

PARA JUGAR:

Después de poner en marcha el programa por la instrucción RUN, se obtiene la visualización de las dos rejillas. Entonces se buscará, por comparación, cuál es el carácter que ha sido modificado (en la rejilla de la derecha).

Después se introduce su número de línea y su número de columna, ambos seguidos de NEWLINE.

Entonces el programa responderá BRAVO o HA PERDIDO.

POSIBLES EXTENSIONES:

Si se dispone de color, será posible utilizar símbolos gráficos de distintos colores, lo que simplificará la búsqueda.

También puede considerarse utilizar dibujos más sofisticados, mejorar el control del resultado, admitir la respuesta buena en la rejilla de la derecha o en la de la izquierda, introducir un cronometraje, etc.

ANOMALÍA

PROGRAMA ZX 81

```
  15 LET T=5
  20 FOR I=SGN PI TO T
  30 PRINT AT NOT PI,I+I;I;TAB I
+I+16;I
  40 PRINT AT I+I,NOT PI;I;TAB 1
6;I
  42 NEXT I
  47 FOR I=SGN PI TO 22
  50 LET L$=CHR$ (11*RND+128*INT
 (RND*2))
  60 LET L=SGN PI
  65 LET S=T
  70 GOSUB 1000
  80 LET L=9
  85 LET S=T+SGN PI
  90 GOSUB 1000
 100 NEXT I
 110 LET S=NOT PI
 120 GOSUB 1000
 130 LET L=CODE L$
 140 INPUT I
 160 INPUT T
 170 PRINT AT I+I,T+T+16;
 180 GOSUB 1010
 185 CLS
 190 PRINT "BRAVO" AND L=C;"HA P
ERDIDO" AND L<>C
 210 STOP
1000 PRINT AT 2*INT (T*RND+SGN P
I),2*INT (S*RND+L);
1010 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
1020 IF C<>CODE "" AND S THEN GO
TO 1000
1030 PRINT L$
1040 RETURN
```

```
  1 2 3 4 5        1 2 3 4 5
1                1
2                2
3                3
4                4
5                5
```

PROGRAMA ZX SPECTRUM

```
  15 LET T=5
  20 FOR I=1 TO T
  30 PRINT AT 0,I+I;I;AT 0,I+I+1
6;I
  40 PRINT AT I+I,0;I;TAB 16;I
  42 NEXT I
  47 FOR I=1 TO 22
  50 LET L$=CHR$ (26*RND+65)
  60 LET L=1
  65 LET S=T
  70 GO SUB 1000
  80 LET L=9
  90 GO SUB 1000
 100 NEXT I
 110 LET S=0
 120 GO SUB 1000
 130 BEEP .3,10: BEEP .3,18: BEE
P .3,8
 140 INPUT "LINEA :";I
 160 INPUT "COLUMNA :";T
 170 PRINT AT I+I,T+T+16;
 180 GO SUB 1010
 185 CLS
 190 PRINT "BRAVO" AND L$=C$;"HA
PERDIDO" AND L$<>C$
 210 STOP
1000 LET U=2*INT (T*RND+1): LET
V=2*INT (S*RND+L)
1010 LET C$=SCREEN$ (U,V)
1020 IF C$<>" " AND S THEN GO T
O 1000
1030 PRINT INVERSE 1; INK INT (4
*RND+1);AT U,V;L$: BEEP .02,0
1040 RETURN
```

```
  1 2 3 4 5        1 2 3 4 5
1                1
2                2
3                3
4                4
5                5
```

Varios en uno

REFERENCIAS: Modalidad: SLOW o FAST.

Dificultad de programación: difícil.

Categoría del juego: para dos jugadores.

Dificultad del juego: variable, según su utilización.

EL JUEGO: En realidad, con este programa pueden realizarse varios juegos; se comportará como un lápiz electrónico, permitiendo registrar los símbolos 0 o X en cualquier lugar de la pantalla del televisor.

Podrá utilizarse, por ejemplo, para el juego de pulgas (gana el primero que consigue una alineación de 5 peones idénticos), para un GO (se recuadran los peones adversarios) o para cualquier otro juego, según la imaginación de cada uno.

EL PROGRAMA: El campo de juego está delimitado por 4 símbolos, situados por la línea 1Ø.

La línea 35 efectúa el cambio de jugador.

El jugador que tiene el turno ve el símbolo que ha escogido (0 o X) situarse en el límite del cuadro (líneas 6Ø y 7Ø).

Las líneas 1ØØ y 11Ø permiten, a los dos jugadores, desplazar el símbolo sobre el campo.

Las líneas 12Ø a 14Ø controlan, si es posible, el desplazamiento (casilla libre y no pasarse de los límites).

La línea 2Ø5 permite escribir en una casilla libre.

PARA JUGAR: RUN provoca la visualización de los límites del campo de juego y del número del jugador, así como los símbolos correspondientes (0 o X), que deberán desplazarse en la pantalla con la ayuda de las 4 teclas de dirección (5, 6, 7, 8).

No se puede ir a una casilla que ya esté ocupada, y será necesario pulsar la tecla Ø para registrar el símbolo propio en una casilla que esté disponible.

Tan pronto como se haya hecho, aparecerá el símbolo del segundo jugador en la columna de la izquierda, casi en el medio, y el jugador actuará de la misma forma que su adversario.

Para salir del programa bastará con pulsar BREAK.

POSIBLES EXTENSIONES: Podrá utilizarse toda la pantalla como campo de juego y realizar controles en función de los distintos juegos que quieran practicarse: control de alineación de 5 piezas, para el juego de la pulga, control del número de piezas capturadas para el GO, etc.

VARIOS EN UNO

PROGRAMA ZX 81

```
  10 PRINT "▛
▜"
  20 PRINT AT CODE "=",NOT PI;"▙
                   ▟"
  30 LET J=NOT PI
  35 LET J=ABS (J-SGN PI)
  40 PRINT AT PI*PI,CODE "*";"JR
. ";J+SGN PI;":O" AND NOT J;":X"
 AND J
  60 LET X=CODE "£"
  70 LET Y=NOT PI
  80 PRINT AT X,Y;"O" AND NOT J;
"X" AND J
  85 PAUSE 4E4
  90 LET A=X
  95 LET B=Y
 100 LET X=X+(INKEY$="6" AND X<C
ODE "(")-(INKEY$="7" AND X>SGN P
I)
 110 LET Y=Y+(INKEY$="8" AND Y<C
ODE "(")-(INKEY$="5" AND Y>SGN P
I)
 115 PRINT AT A,B;" "
 120 PRINT AT X,Y;
 130 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
 140 IF C<>CODE "O" AND C<>CODE
"X" THEN GOTO CODE "COS "
 150 PRINT CHR$ C
 160 LET X=A
 170 LET Y=B
 200 IF INKEY$<>"0" THEN GOTO 80
 205 PRINT "O" AND NOT J;"X" AND
 J
 210 GOTO CODE "7"
```

```
▛                    ▜

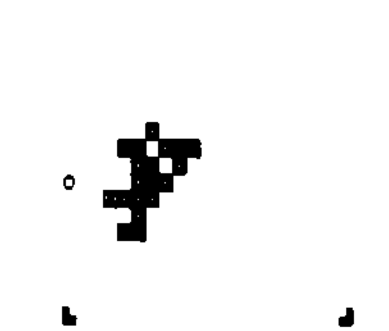


                    JR. 1:0
        OOO
O       OXX
         X
        X
         X


▙                    ▟
```

PROGRAMA ZX SPECTRUM

```
  10 PRINT "▛";TAB 20;"▜",AT 20,
0;"▙";TAB 20;"▟"
  30 LET J=0
  35 LET J=ABS (J-1)
  60 LET X=12
  70 LET Y=0
  80 PRINT AT X,Y;"O" AND NOT J;
"X" AND J
  85 PAUSE 4E4
  90 LET A=X
  95 LET B=Y
 100 LET X=X+(INKEY$="6" AND X<1
9)-(INKEY$="7" AND X>1)
 110 LET Y=Y+(INKEY$="8" AND Y<1
9)-(INKEY$="5" AND Y>1)
 115 PRINT AT A,B;" "
 120 PRINT AT X,Y;
 130 LET C$=SCREEN$ (X,Y)
 140 IF C$<>"O" AND C$<>"X" THEN
 GO TO 200
 150 PRINT INVERSE 1; BRIGHT 1;
INK 1+(C$="X");C$
 160 LET X=A
 170 LET Y=B
 200 IF INKEY$<>"0" OR Y=0 THEN
GO TO 80
 205 PRINT INVERSE 1; BRIGHT 1;
INK 1+J;"O" AND NOT J;"X" AND J:
 BEEP .03,10
 210 GO TO 35
```



Recorrido del caballo

REFERENCIAS:

Modalidad: SLOW.

Dificultad de programación: difícil.

Categoría del juego: de reflexión.

Dificultad del juego: difícil.

EL JUEGO:

Se presenta aquí como un juego completamente de azar, aunque en realidad proviene de un problema bastante difícil de resolver: cómo desplazar un caballo en un tablero de ajedrez de forma que pase una sola vez por cada una de las 64 casillas del tablero.

El caballo se desplaza siguiendo las reglas comúnmente admitidas en el ajedrez, y su posición de salida está definida por el jugador; por el contrario, sus desplazamientos se efectúan al azar. ¿Logrará recorrer las 64 casillas?

EL PROGRAMA:

Las líneas 15 y 2Ø definen, en dos variables alfanuméricas A$ y B$, los distintos desplazamientos posibles; los dos bucles intercalados I (25-45) y J (3Ø-4Ø) dibujan el campo de juego.

Las coordenzdas de la casilla de salida se introducen en las líneas 5Ø y 55 (línea y después columna).

La línea 66 imprime el número de la jugada.

La línea 14Ø permite la tirada al azar de un desplazamiento del caballo.

El bucle Z (145-17Ø) busca si este desplazamiento es posible; si no lo es, se probarán todos los otros (siempre en el mismo orden).

Las líneas 146 y 147 efectúan el desplazamiento.

La línea 148 controla que no se salga del campo de juego.

Las líneas 15Ø y 16Ø bifurcan a la línea 2ØØ, si la jugada es válida.

La línea 18Ø gestiona el final de la partida (caballo bloqueado) y visualiza la puntuación.

Las líneas 2ØØ, 21Ø y 22Ø realizan el golpe e incrementan la puntuación.

PARA JUGAR:

Se pone en marcha el programa por la orden RUN.

Se visualiza el campo de juego; se escoge entonces la posición de salida, tecleando primero un número de línea (entre 1 y 8), seguido de NEWLINE, y después un número de columna, seguido de NEWLINE.

El juego empieza inmediatamente y se desarrolla de forma completamente automática hasta que el caballo quede bloqueado. Naturalmente, se puede parar en cualquier momento pulsando BREAK.

POSIBLES EXTENSIONES: Pueden imaginarse numerosas variantes sobre este tema, aunque sólo sea empezar por variar el tamaño de los tableros de ajedrez; también puede mejorarse considerablemente el grafismo o utilizar color. Pero lo más interesante es lograr definir un algoritmo, dando la o las soluciones del problema inicial.

RECORRIDO DEL CABALLO

PROGRAMA ZX 81

```
  10 LET S=SGN PI
  15 LET A$="56653223"
  20 LET B$="653223"+A$+"56"
  25 FOR I=S TO LEN A$
  30 FOR J=S TO LEN A$
  35 PRINT AT I+I,INT PI*J;"."
  40 NEXT J
  45 NEXT I
  50 INPUT I
  55 INPUT J
  60 LET A$=A$+A$
  66 PRINT AT I+I,INT PI*J;S
 140 LET Z=INT (CODE "■"*RND+SGN
 PI)
 145 FOR Z=Z TO Z+CODE "■"
 146 LET A=I+VAL A$(Z)-CODE "■"
 147 LET B=J+VAL B$(Z)-CODE "■"
 148 IF A<SGN PI OR B<SGN PI OR
A>CODE "■" OR B>CODE "■" THEN NE
XT Z
 150 PRINT AT A+A,INT PI*B;
 160 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "." THEN GOTO CODE
 "COS "
 170 NEXT Z
 180 PRINT AT CODE "=",NOT PI;S;
U
 200 LET I=A
 210 LET J=B
 220 LET S=S+SGN PI
 230 GOTO CODE "PI"
```

```
 .  .  23 2  27 4  15 46
22  .  .  .  14 1  26 5
 .  .  13 24 3  28 45 16
 .  21 .  .  .  25 6  29
 .  12 19 40 35 44 17 42
20 37 34  .  18 41 30 7
11  .  39 36 9  32 43 .
38  .  10 33 .  .  8  31

46
```

PROGRAMA ZX SPECTRUM

```
  10 LET S=1
  15 LET A$="56653223"
  20 LET B$="653223"+A$+"56"
  25 FOR I=S TO LEN A$
  30 FOR J=S TO LEN A$
  35 PRINT PAPER 5;AT 0,3*I;I;AT
 I+I,0;I; PAPER 2;AT I+I,3*J;"."
  40 NEXT J
  45 NEXT I
  50 INPUT "LINEA DE SALIDA :";I
  55 INPUT "COLUMNA DE SALIDA :"
;J
  60 LET A$=A$+A$
  66 PRINT INVERSE 1; INK 4;AT I
+I,3*J;S
 140 LET Z=INT (8*RND+1)
 145 FOR Z=Z TO Z+8
 146 LET A=I+VAL A$(Z)-4
 147 LET B=J+VAL B$(Z)-4
 148 IF A<1 OR B<1 OR A>8 OR B>8
 THEN NEXT Z
 160 IF SCREEN$ (A+A,3*B)="." TH
EN GO TO 200
 170 NEXT Z
 180 PRINT FLASH 1;AT 20,0;"TOTA
L :";S: BEEP .3,10: BEEP .3,16:
BEEP .3,8: STOP
 200 LET I=A
 210 LET J=B
 220 LET S=S+1: BEEP 0.1,13
 230 GO TO 66
```

```
   1  2  3  4  5  6  7  8

1  ■  .  ■  ■  ■  ■  .  .
2  ■  ■  ■  ■  ■  .  ■  ■
3  .  ■  ■  ■  ■  ■  ■  .
4  .  ■  ■  .  ■  ■  ■  ■
5  ■  .  ■  ■  ■  ■  ■  ■
6  .  .  ■  ■  ■  ■  ■  ■
7  ■  ■  .  ■  ■  .  .  .
8  .  .  ■  .  .  ■  ■  .

TOTAL :44
```

Pared de ladrillos

REFERENCIAS: Modalidad: SLOW.
Dificultad de programación: difícil.
Categoría del juego: reflejos (para un jugador).
Dificultad del juego: difícil.

EL JUEGO: Si existe un juego que pueda presentarse como un gran clásico de los juegos de arcada para microordenador es, sin duda alguna, la pared de ladrillos (en inglés BREAKOUT).

Aquí se ofrece una versión 1K, aunque, sin embargo, ofrece buenos resultados: visualización de la puntuación al final de la partida, rebotes aleatorios, etc.

La regla es muy elemental: se dispone de una raqueta que se desplaza horizontalmente para enviar una pelota (materializada por un punto) que derribará los ladrillos dibujados en la parte superior de la pantalla.

El juego consiste en derribar el mayor número posible de ladrillos sin fallar con la pelota, lo que provocaría el final de la partida y la visualización de la puntuación.

EL PROGRAMA: El bucle I (3Ø-5Ø) dibuja el campo de juego y los ladrillos.

La línea 55 dibuja la raqueta, cuya posición se controla por la línea 6Ø.

Las líneas 85, 9Ø y 1ØØ miran si se ha tocado la pared, los ladrillos o la raqueta.

Los rebotes se administrarán, en consecuencia, con cálculos aleatorios por las líneas 132 y 138 (el matiz gris empleado corresponde aquí al *graphic shift H*).

La línea 64 visualiza la puntuación al final de la partida.

PARA JUGAR: El programa se pondrá en marcha por la instrucción RUN, lo que provoca la visualización del campo de juego, y después, muy rápidamente, el envío de la pelota.

El jugador intentará desplazar su raqueta con la ayuda de las teclas de desplazamiento horizontal (5 y 8).

Si, en un momento dado, la pelota no toca la raqueta, se visualizará la puntuación y la partida terminará.

Se puede acabar en cualquier momento pulsando BREAK.

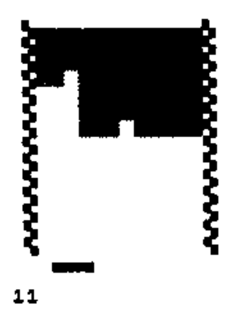
POSIBLES EXTENSIONES: Puede considerarse ampliar el campo de juego, aumentar el número de ladrillos, prever varias pelotas, dibujar cada hilera de ladrillos de un color diferente, introducir efectos sonoros, etc.

Inspirándose en los juegos de arcada, también podrán introducirse pequeños obreros que reconstruirán el muro de ladrillos a medida que se vaya derribando.

PARED DE LADRILLOS

PROGRAMA ZX 81

```
   5 LET S=NOT PI
  10 LET Y=PI+PI
  20 LET J=Y
  30 FOR I=NOT PI TO CODE "$"
  40 PRINT TAB SGN PI;"███████████
█████" AND NOT I;"███████████████
" AND I AND I<Y;"█              █"
 AND I>Y
  50 NEXT I
  52 LET H=-SGN PI
  54 LET K=H
  55 PRINT AT CODE ":",Y;" ███ "
  60 LET Y=Y+(INKEY$="8" AND Y<C
ODE "£")-(INKEY$="5" AND Y>SGN P
I)
  62 PRINT AT I,J;" "
  64 IF I>CODE ":" THEN PRINT S;
W
  66 LET I=I+H
  80 LET J=J+K
  85 PRINT AT I,J;
  90 LET C=PEEK (PEEK 16396+256*
PEEK 16399)
 100 IF C THEN GOTO C
 118 PRINT "▘"
 120 GOTO CODE "R"
 131 LET H=-H
 132 LET K=RND-RND
 134 LET K=-K
 135 GOTO CODE "PI"
 136 LET S=S+SGN PI
 138 LET K=-(K+(2*(RND<.5)-SGN P
I)*RND/INT PI)
 140 LET H=-H
 142 GOTO CODE "R"
```



PROGRAMA ZX SPECTRUM

```
   5 LET S=0
  10 LET Y=6
  20 LET J=Y
  30 FOR I=0 TO 16
  40 PRINT TAB 1; PAPER 2;"|_____
_______|" AND NOT I; PAPER 2;"|
" AND I AND I<=Y; INVERSE 1; INK
 3+INT (I/2.5);"##############" AN
D I AND I<=Y; INVERSE 0; INK 0;
PAPER 2;"|" AND I AND I<=Y;"|" A
ND I>Y; PAPER 7;"              " A
ND I>Y; PAPER 2;"|" AND I>Y
  50 NEXT I
  52 LET H=-1
  54 LET K=H
  55 PRINT INVERSE 1; INK 1;AT 1
7,Y;"█▁▁█"
  60 LET Y=Y+(INKEY$="8" AND Y<1
2)-(INKEY$="5" AND Y>1)
  62 PRINT AT I,J;" "
  64 PRINT AT 20,17;"MARCADOR :"
;S: IF I>20 THEN BEEP 1,0: PAUSE
 4E4: RUN
  66 LET I=I+H
  80 LET J=J+K
  85 PRINT AT I,J;
  90 LET C$=SCREEN$ (I,J)
 100 IF C$<>" " THEN BEEP .03,0:
 GO TO 131+3*(C$="|")+5*(C$="#")
 118 PRINT "O"
 120 GO TO 55
 131 LET H=-H
 132 LET K=RND-RND
 134 LET K=-K
 135 GO TO 66
 136 LET S=S+1
 138 LET K=-(K+(2*(RND<.5)-1)*RN
D/3)
 140 LET H=-H
 142 GO TO 55
```

MARCADOR :26

Othello

REFERENCIAS:

Modalidad: SLOW o FAST.

Dificultad de programación: difícil.

Categoría del juego: juego de reflexión (para dos jugadores).

Dificultad del juego: difícil.

EL JUEGO:

Convertido en un gran clásico de los juegos sobre microordenador, el juego OTHELLO o REVESINO mejora al gestionarse sobre ordenador con visualización, aunque aquí se practique con dos jugadores; el juego consiste esencialmente en dar la vuelta a los peones del adversario; esto puede parecer, a la larga, muy aburrido, sobre todo si se es un asiduo jugador.

El principio consiste en que el jugador que posee las X debe, para poder jugar, rodear una o varias piezas del adversario en una o varias direcciones; si lo consigue, se dará entonces la vuelta a las O, que se convertirán en X (la vuelta podrá hacerse horizontal o verticalmente, o siguiendo las diagonales).

El jugador que posea las O procederá de la misma manera, y así sucesivamente hasta que el tablero esté lleno o nadie pueda jugar más (si un jugador no puede dar la vuelta a los peones, perderá su turno).

EL PROGRAMA:

Como ha sido difícil poner el programa sobre 1K octeto, ha sido necesario sacrificar algo la presentación.

El bucle P (11-26) dibuja el campo de juego y visualiza una X o una O, dependiendo del jugador que comienza.

El bucle P (41-212) controla la validez de la jugada (¿está la casilla libre? : líneas 2Ø3 a 2Ø9), utilizando una parte del programa que se encuentra en las líneas 224 a 242 (se han cogido uno o varios enemigos en la línea 224 y eventual vuelta autorizada por la línea 23Ø: se vuelve a explorar en todas las direcciones si el rodeo es posible).

La línea 41 define las 8 direcciones de exploración.

Las líneas 56 y 59 calculan los desplazamientos que deben efectuarse para explorar estas 8 direcciones.

Las líneas 66 y 2ØØ realizan los desplazamientos.

La línea 218 asegura el cambio de jugador cuando hay vuelta (se controla en la línea 215).

PARA JUGAR:

Después de RUN, cada jugador introducirá por turnos las coordenadas (línea, NEWLINE, y después columna, NEWLINE) de la casilla que desea jugar; si se acepta la jugada, se dará la vuelta a los peones y se hará el cambio de jugador; si la jugada no es posible, se introducirá Ø como primera coordenada, con el fin de realizar el cambio de jugador.

POSIBLES EXTENSIONES: Ante todo, por supuesto, mejorar la visualización con colores, un tablero más aireado, etc.

También podrá imaginarse un juego de máquina para el ZX.

Nota: Después de haber puesto en marcha el programa por RUN, será conveniente introducir las dimensiones del tablero (3, NEWLINE, o 4, ...).

OTHELLO

PROGRAMA ZX 81

```
   5 INPUT A
   8 LET B=SGN PI
  11 FOR P=B TO A+A
  14 PRINT AT NOT PI,P;P;AT P,NO
T PI;P
  17 NEXT P
  20 PRINT AT A,A;"XO";AT A+B,A;
"OX"
  23 LET K=B
  26 PRINT AT A*A,NOT PI;"X" AND
 K;"O" AND NOT K
  29 INPUT U
  32 IF NOT U THEN GO TO CODE "
AND "
  35 INPUT V
  38 LET H=PI
  41 FOR P=B TO VAL "8"
  44 LET S=NOT PI
  47 LET W=NOT PI
  50 LET T=U
  53 LET R=V
  56 LET F=(P<INT PI OR P>PI+PI+
B)-(P>=PI AND P<=PI+PI)
  59 LET G=(P>=INT PI+INT PI)-(P
>B AND P<=PI+B)
  62 LET R$=("X" AND S=B)+("O" A
ND S=B+B)
  64 PRINT AT T,R;R$
  66 LET T=T+F
 200 LET R=R+G
 203 PRINT AT T,R;
 206 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
 209 IF C=CODE "O" OR C=CODE "X"
 THEN GO TO CODE " STEP "
 212 NEXT P
 215 IF H THEN GO TO CODE ";"
 218 LET K=ABS (K-B)
 221 GO TO CODE ";"
 224 IF (C=CODE "O" AND K) OR (C
=CODE "X" AND NOT K) THEN LET W=
B
 227 PRINT R$
 230 IF W AND ((C=CODE "O" AND K
) OR (C=CODE "X" AND NOT K)) THE
N GO TO CODE "PI"
 233 IF NOT W OR S THEN GO TO CO
DE "USR "
 236 LET S=B+NOT K
 239 LET H=H*NOT S
 242 GO TO CODE "J"
```

```
 12345678
1
2
3    XO
4   XXO
5   OXO
6     XX
7      X
8




O
```

PROGRAMA ZX SPECTRUM

```
   5 INPUT "DIMENSIONES :";A
   8 LET B=1
  11 FOR P=B TO A+A
  14 PRINT AT 0,P+P;P;AT P+P,0;P
  17 NEXT P
  20 PRINT INVERSE 1; INK 2;AT A
+A,A+A;"X";AT A+A,A+A+B+B; INK 5
;"O";AT A+A+B+B,A+A;"O"; INK 2;A
T A+A+B+B,A+A+B+B;"X"
  23 LET K=B
  26 PRINT INVERSE 1; FLASH 1; I
NK 2+3*NOT K;AT 21,0;"X" AND K;"
O" AND NOT K
  29 INPUT "LINEA :";U
  32 IF NOT U THEN GO TO 218
  35 INPUT "COLUMNA :";V
  38 LET H=PI
  41 FOR P=B TO 8
  44 LET S=0
  47 LET W=0
  50 LET T=U+U
  53 LET R=V+V
  56 LET F=(P<3 OR P>7)-(P>3 AND
 P<7)
  59 LET G=(P>=6)-(P>B AND P<5)
  62 LET R$=("X" AND S=B)+("O" A
ND S=B+B)
  64 PRINT INVERSE 1; INK 2+3*NO
T K;AT T,R;R$
  66 LET T=T+F+F
 200 LET R=R+G+G
 203 PRINT AT T,R;
 206 LET C=CODE SCREEN$ (T,R)
 209 IF C=CODE "O" OR C=CODE "X"
 THEN GO TO 224
 212 NEXT P
 215 IF H THEN GO TO 26
 218 LET K=ABS (K-B): BEEP .4,10
 221 GO TO 26
 224 IF (C=CODE "O" AND K) OR (C
=CODE "X" AND NOT K) THEN LET W=
B
 227 PRINT INVERSE 1; INK 2+3*NO
T K;R$
 230 IF W AND ((C=CODE "O" AND K
) OR (C=CODE "X" AND NOT K)) THE
N GO TO 66
 233 IF NOT W OR S THEN GO TO 21
2
 236 LET S=B+NOT K
 239 LET H=H*NOT S
 242 GO TO 47
```

```
   1 2 3 4 5 6 7 8

1

2

3          ■ ■

4        ■ ■ ■

5        ■ ■ ■

6            ■ ■

7            ■ ■

8          ■ ■ ■ ■

■
```

Anexo



LA ORGANIZACIÓN DE LA MEMORIA EN EL ZX 81

Rutinas USR
RAMTOP
Gestión de los GOSUB
ERR-SP
Pila ordenadora
SP
Reserva
STREND
Pila calculadora
STKBOT
Líneas en entradas + espacio de trabajo
E-LINE
Variables
VARS
Archivo de visualización
Programa
16509
Variables del sistema
16384

RAM utilizable por el programa

direcciones de los diversos apuntadores

LA ORGANIZACIÓN DE LA MEMORIA EN EL ZX SPECTRUM

Rutinas USR
RAMTOP
Gestión de los GOSUB
D-FILE
Pila ordenadora
SP
Reserva
STREND
Pila calculadora
STKBOT
Espacio de trabajo
Datos de entrada
WORK SP
Líneas en entradas
E-LINE
VARS
PROG
Información de canal
CHANS
Microdrive
23734
Variables del sistema
23552
Memoria intermedia de la impresora
23296
Archivo de atributos
22528
Archivo de visualización
16384

RAM utilizable por el programa

direcciones de los diversos apuntadores